University of Canterbury
Department of Mathematics and Statistics

Rekernelisation Algorithms in Hybrid Phylogenies

**A thesis submitted in
partial fulfilment
of the requirements of
the Degree for
Master of Science in Mathematics
at the
University of Canterbury
by**

Joshua Stewart Collins

Supervisor: Assoc. Prof. Charles Semple

2009

Abstract

It has become well known that an evolutionary tree is inadequate to represent fully the history of life. Two possible ways of dealing with this are the rooted subtree prune and regraft distance between a pair of trees, which measures how different they are, and the slightly more biologically sound hybridisation number of a set of trees that attempts to determine the minimum number of hybrid events that must have occurred for a given set of evolutionary trees. When characterised via agreement forests both problems are, although NP hard, fixed parameter tractable—meaning the problem can be converted to a similar problem with a smaller input size.

This thesis investigates ways of improving existing algorithms for calculating the minimum rooted subtree prune and regraft distance and hybridisation number for a pair or, in the latter case, set of trees. In both cases a technique is used that allows the problem to be rekernelised during the run of the program. Another, less effective method, is also looked at which finds the rooted subtree prune and regraft distance or hybridisation number solely on what cannot be contained within any agreement forest.

Additionally the characterisation of the minimum rooted subtree prune and regraft distance via maximum agreement forests is extended to non-binary trees and the hybridisation number of a set of phylogenetic trees is extended to unrooted trees.

Implementation Details The software was run on a PC with a 2.66GHz processor and 2G RAM. All code, except the pre-existing PERL code, was written in Java.

Acknowledgements First and foremost to my supervisor Charles Semple. Next to the person whose suggestions kick started many of the results in this thesis Simone Linz. Also to the people who helped maintain my sanity during the year: my mum, my office mates Tim Candy and Michael Snook and my good friends André Geldenhuis, Storm Geldenhuis and Rachel Bradstock.

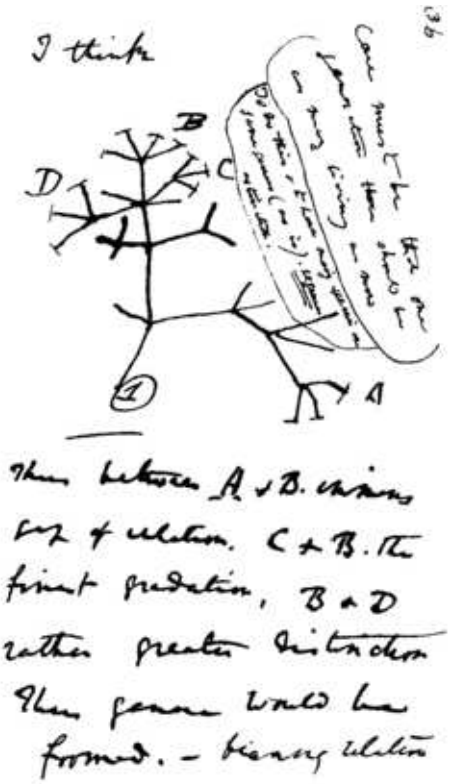


Figure 1: Arguably the first ever evolutionary tree, from Charles Darwin's *B* notebook.

Contents

Chapter 1

Preliminaries

> "It is difficult to overemphasise the importance of hybridisation and polyploidy in evolution."
>
> Funk (1985)

> "Molecular phylogeneticists will have failed to find the 'true tree', not because their methods are inadequate or because they have chosen the wrong genes, but because the history of life cannot properly be represented as a tree."
>
> Doolittle (1999)

1.1 Introduction

Although the history of evolutionary theory has its roots amongst the ancient Greeks, Romans, Chinese and Muslims[1] it was not until 1837 that these ideas culminated in the first sketch of an evolutionary tree in Charles Darwin's *B* notebook[2]. Upon the publishing of Charles Darwin's Origin of Species (Darwin, 1859) it quickly became thought that such trees, like the stylised one in Figure 1.1, would be suitable to explain the descent of all species. However it has become apparent that such trees, although invaluable, are not always the most appropriate representation of evolution. Events such as hybridisation, horizontal gene transfer, recombination and endosymbiosis give rise to a collection of events known as *reticulation events* that confound the tree model of evolution. For although certain groups, such as mammals, have experienced little

[1] `http://en.wikipedia.org/wiki/History_of_evolutionary_thought`

[2] See Figure 1.

reticulation it has played a major part in the evolution of certain plant, fish and bacterial groups. (Posada and Crandall, 2001)

If biological concerns were not sufficient motivation for new methods there are other fields in which evolutionary theory is utilised. In linguistics, languages may diverge from one another much as biological species do. It is no surprise that languages change over time and that two isolated languages will change differently, in time becoming mutually unintelligible. However, borrowing of sounds, words and grammatical ideas from other languages, which occurs more frequently than one might think, do not fit into a tree model of language evolution. (Nakhleh, Ringe, and Warnow, 2002)

Reticulation type events also arises in textual philology in which the aim is to reconstruct an author's original text based on the copies that still survive. Naturally the copies are not error-free, particularly those made by hand, and so a process easily modelled by evolution arises. Moreover, equivalent events to reticulation also exist in which past philologists have attempted to culminate all known texts into one that is a better match to the original, only to have introduced their own errors or have errors introduced by subsequent copiers of their work. (Spencer, Davidson, Barbrook, and Howe, 2004)

The question then becomes how to best incorporate these events. The tree used to describe the descent of species is characterised by a labelling on the leaves corresponding to extant species, internal vertices corresponding to hypothetical ancestral species and edges denoting the transfer of genetic material. Intuitively reticulation could correspond to taking a piece of one tree which displays one line of descent, detaching then reattaching it elsewhere in order to obtain another tree with the other expressed line of descent, leading to tree metrics such as nearest neighbour interchange, subtree prune and regraft and tree bisection and reconnection and the related rooted equivalents. Given such metrics there are then algorithms that attempt to locate the "best" phylogenetic tree for a set of data. (Baroni, Semple, and Steel, 2004; Beiko and Hamilton, 2006; Hein, 1990; Hein, Jiang, Wang, and Zhang, 1996; Maddison, 1991, 1997; McFadden and Gilson, 1995; Nakhleh, Warnow, Linder, and St. John, 2005; Song and Hein, 2003)

Another alternative to model the situation is a directed graph whose components carry much the same connotations as the previously described tree. (Baroni et al., 2004; Bryant and Moulton, 2004; Nakhleh et al., 2005) One approach is to build a tree and use a heuristic to add edges (Legendre and Makarenkov, 2002), but there are a number of theoretical results from mathe-

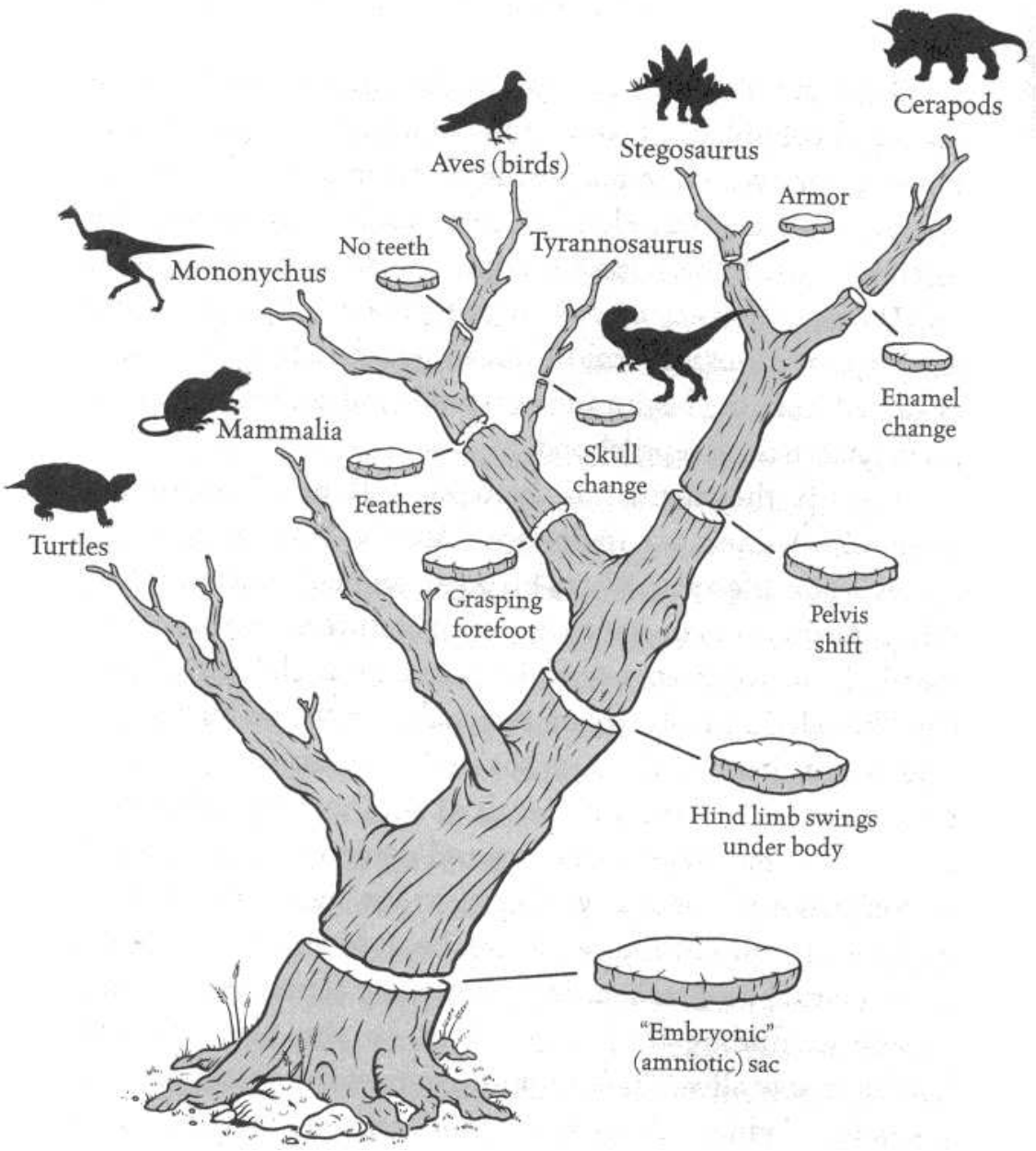


Figure 1.1: A phylogenetic tree for Amniotes (source unknown).

maticians and computer scientists that attempt a more accurate solution. (Baroni et al., 2004; Gusfield and Bansal, 2005; Gusfield, Eddhu, and Langley, 2004; Holland, Huber, Moulton, and Lockhart, 2004; Huson, Dezulian, Kloepper, and Steel, 2004; Huson, Kloepper, Lockhart, and Steel, 2005; Moret, Nakhleh, Warnow, Linder, Tholse, Padolina, Sun, and Timme, 2004) Unfortunately some of the problems that have arisen out of these new approaches are computationally hard. Fortunately, some of these problems, in particular the ones this thesis is concerned with, exhibit a property known as fixed parameter tractability and through this "fast" algorithms have been devised. (Allen and Steel, 2001; Bordewich and Semple, 2004, 2007b; Bordewich, Linz, St. John, and Semple, 2007a; Linz and Semple, 2008)

In this thesis improved fixed-parameter algorithms are provided for the following problems from phylogenetic studies: calculating the minimum number of rooted subtree prune and regraft operations required to transform one tree into another, and the minimum number of reticulation events required for a set of trees. In addition to this rooted subtree prune and regraft is extended to non-binary trees, the notion of hybridisation number is extended to unrooted trees, and a chain reduction for the non-binary hybridisation number that extends better to more than a pair of trees is developed. A number of algorithms are also implemented and tested on real world data, namely the grass data set from Grass Phylogeny Working Group (2001).

1.2 Organisation

This thesis is organised as follows: Chapter 2 contains the notation and definitions that are used throughout this thesis; the reader familiar with the definitions used in Semple and Steel (2003) may omit this without much detriment. Chapter 3 deals with rooted subtree prune and regraft results. The first section in this chapter contains definitions and previous results, and a characterisation of rooted subtree prune and regraft for non-binary phylogenetic trees. Then a chain reduction is proved to show the problem is fixed parameter tractable. The next section returns to binary phylogenetic trees and details a method by which the rSPR-EXACT bounded search—whose pseudo-code is first given in Bordewich, McCartin, and Semple (2007b)—may be improved by allowing possible rekernelisation during the run of the algorithm. This modified algorithm is run on the *poaceae* data set from Grass Phylogeny Working Group (2001) and compared to an implementation of the rSPR-EXACT algorithm.

Chapter 4 makes up the rest of the thesis and deals with phylogenetic networks and the

hybridisation number problem. After some introductory comments, the first algorithm devised looks at what characterises agreement forests for a set of trees by determining what they may not contain. Some further comments are made about the algorithm that allow paths in the search tree to be ignored, since they will not give better than the optimal results, and pseudo-code for the algorithm is given. The subsequent sections 4.3–4.4 deal with an algorithm that, like the one in the previous paragraph, allows the hybridisation number problem to be potentially rekernelised during the run of the algorithm. This method leads to a non-binary chain reduction that scales better when dealing with multiple trees. Pseudo-code is given and both binary algorithms are run on the *poaceae* data set along with the PERL program written and tested in Bordewich et al. (2007a).

The final section in Chapter 4 formulates a way that allows the concept of acyclic agreement forests to be formulated for sets of phylogenetic trees, some or all of which may be unrooted. Although subtree and chain reductions both exist in this new scenario, the absence of a direction on the edges means the pre-existing chain reductions may not be applied, and so it is not clear if this characterisation is fixed parameter tractable.

Chapter 5 then discusses future goals that I believe would be worthwhile pursuing. A bibliography, list of figures and index then follow. The first two chapters and sections titled *A Brief Introduction to...* do not contain original results. Otherwise all the work is original with exceptions noted. The material on rekernelising in order to calculate the hybridisation number (the majority of section 4.3) has been submitted as Collins, Linz, and Semple (submitted).

Chapter 2

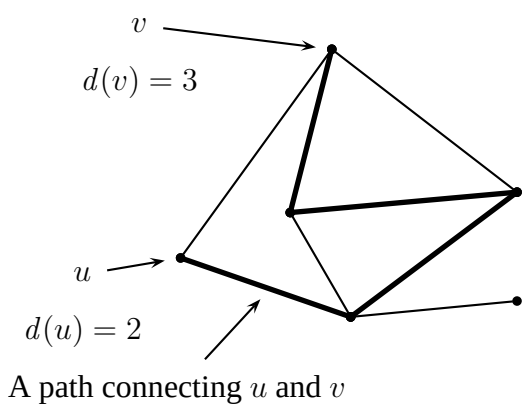
Definitions

This chapter contains definitions that shall be used throughout this thesis. Following in Leibniz' footsteps an attempt has been made to use notation that reflects use, whilst also avoiding creating new impediments to understanding by creating new symbols. As such the notation follows no other text exactly but follows Semple and Steel (2003) closely.

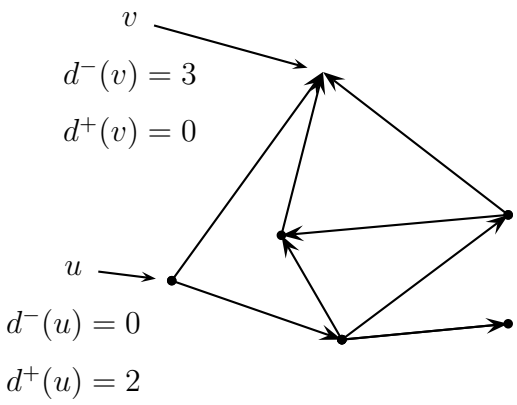
2.1 Graphs and Digraphs

A *graph* G is an ordered pair of sets. The non-empty set of *vertices* of G denoted $V(G)$ and the set of *edges* of G denoted $E(G)$ such that $E(G) \subseteq \{\{x, y\} : x, y \in V(G)\}$. A *directed graph* or *digraph* D is an ordered pair of a set of vertices and a set of *arcs* $A(D)$ which is a collection of ordered pairs $A(G) \subseteq \{(x, y) : x, y \in V(G)\}$. An edge or arc connecting two vertices is said to be *adjacent* to the vertices. Additionally if (u, v) is an arc then v is the *head* of the arc and u is the *tail*. The *(underlying) edge set* of a digraph, also $E(D)$, is the set $\{\{x, y\} : (x, y) \in A(D)\}$. In figures vertices will be represented by points and the vertices u and v will be connected by lines if and only if $\{u, v\}$ is in the edge set of the graph. In the case of a directed graph if (u, v) is in the arc set then an arrow from u to v will connect the two vertices, with the exceptions for phylogenetic trees and networks as noted in the next section.

A graph is *connected* if there is a path in the (underlying) edge set from every vertex in the graph to every other vertex. In a figure this corresponds to never being able to separate one graph into two pieces with no edges between them. A graph is *simple* if it contains no edges from a vertex to itself, and no more than one edge connects each pair of vertices. For practical purposes all of the graphs in this thesis may be considered to be simple. Two simple graphs G



(a) A graph G.



(b) An acyclic digraph such that the underlying edge set gives a graph that is isomorphic to G.

Figure 2.1: A basic connected simple graph and acyclic digraph.

---

and G' are *isomorphic* $G \cong G'$ if there is a bijection ϕ such that $\{u, v\} \in E(G)$ if and only if $\{\phi(u), \phi(v)\} \in E(G')$, or in the arc set in the case of a digraph, and if v is labelled then $\phi(v)$ has the same label otherwise $\phi(v)$ is unlabelled. Given two images of graphs they are isomorphic if one can shift around the vertices until they "look" the same.

A *walk* is a sequence of vertices $v_0, v_1, \ldots, v_k$ such that each $\{v_i, v_{i+1}\}$ is in $E(G)$ or each (v_i, v_{i+1}) is in $A(G)$, in which case it is a *directed walk*, for each $i \in \{0, 1, \ldots, k-1\}$. This corresponds to an unbroken line in the graph that starts at v_0 and passes through each v_i until reaching v_k. In the case of a digraph one proceeds from tail to head of the arrow instead of simply following the edges. A *path* is a walk with every v_i for $i \in \{0, 1, \ldots, k\}$ distinct. In an image this is the same as a walk except that no vertex is ever crossed twice. If there is a directed path from vertex u to vertex v then this partial ordering will be denoted $u \rightsquigarrow v$ whereas an undirected path gives the equivalence relation $u \sim v$. A *(directed) cycle* is a (directed) path with $v_0 = v_k$. That is a path that finishes where it begins. A (di)graph is *acyclic* if it contains no (directed) cycles.

The *degree* of a vertex v, denoted $d(v)$, is the number of edges incident with v. The *out-degree* $d^+(v)$ is the number of adjacent arcs with their first component in v and similarly the *in-degree* $d^-(v)$ is the number of adjacent arcs with their second component in v. A digraph is *rooted* if there is a vertex ρ called the *root* such that $d^-(\rho) = 0$ and there is a directed path from ρ to every vertex of the digraph. A digraph that is not rooted is *unrooted*. For technical reasons

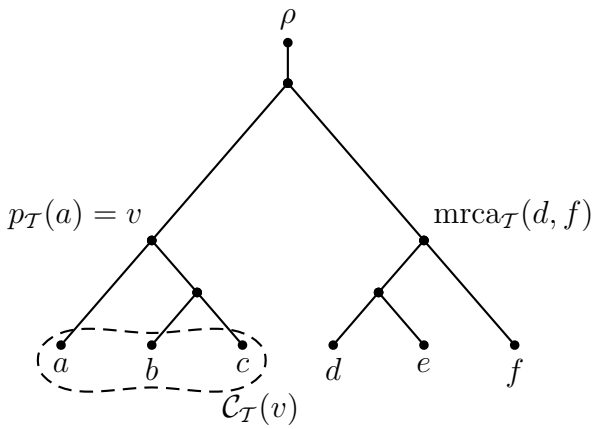


Figure 2.2: A rooted binary phylogenetic tree $\mathcal{T}$ with label set $\mathcal{L}(\mathcal{T}) = \{a, b, c, d, e, f\}$.

---

it becomes advantageous to extend the root out on an edge of its own, thus consider all rooted graphs in this thesis to have a root with such a property.

An *edge contraction* is an operation that identifies two vertices adjacent to some common edge and removes the edge from the edge set. A *contraction of the edge set of a digraph* is an operation either that deletes any vertex v with both $d^-(v) = 1$ and $d^+(v) = 1$ and replaces the edges adjacent to v, say $\{u, v\}, \{v, w\}$ with a single edge $\{u, w\}$ or deletes any unlabelled vertex v with $d^-(v) = 1$ and $d^+(v) = 0$ and deletes its adjacent edge. The motivation for edge contractions is largely due to being interested in how certain species are related to each other. An unlabeled vertex with no descendants corresponds to a hypothetical animal whose descendants, if any, include none of the species under consideration. In the other case we have two hypothetical ancestral species with the same set of descendant species and so gain no useful information from keeping both in the tree.

2.2 Trees

A *tree* $\mathcal{T}$ is a connected (di)graph with no cycles in the underlying edge set. If a vertex v of $\mathcal{T}$ has degree one and is not the root, unless the root is the only vertex in the tree, then v is called a *leaf*, otherwise v is an *internal vertex*. A *binary tree* $\mathcal{B}$ is a tree in which every internal vertex has degree three except the root, if it exists which has degree one. A *binary refinement* $\mathcal{B}$ of a phylogenetic tree $\mathcal{T}$ is a binary tree such that there is some contraction of edges of $\mathcal{B}$ that gives a tree isomorphic to $\mathcal{T}$. In context of phylogenetic trees it means all the evolutionary

relationships expressed in $\mathcal{T}$ are expressed in $\mathcal{B}$. For an example see Figure 3.5.

A *phylogenetic tree with label set* X is a tree with leaves labeled by distinct members of X. Labelled leaves will be identified with their labels. The *label set* of a phylogenetic tree $\mathcal{T}$ denoted $\mathcal{L}(\mathcal{T})$ is the set of labels and, if $\mathcal{T}$ is rooted, the root denoted ρ. Additionally we will assume a total ordering on the labels of the leaves. As the phylogenetic tree is used to model evolution when rooted the edges admit a natural direction, namely away from the root. When drawn this direction will be assumed to proceed down the page and not depicted. Moreover phrases such as *the edge set of a phylogenetic tree* $\mathcal{T}$ will frequently be used in which case the underlying edge set is being referred to and in general the terms arc set and edge set will be used interchangeably. If there is a directed path $u \rightsquigarrow v$ for $u, v \in V(\mathcal{T})$ of a phylogenetic tree $\mathcal{T}$ then u is said to be an *ancestor* of v and v is a *descendant* of u. If (u, v) is an arc then u is the *parent* of v denoted $p(v) = u$ and v is a *child* of u. When it is necessary to specify the tree $\mathcal{T}$ in which we are considering the parent we write $p_{\mathcal{T}}(v) = u$. A *cherry* of a tree is a pair of leaves in the tree that share the same parent. Given a set $A \subseteq X$ for a phylogenetic tree $\mathcal{T}$ with leaves labelled by X the *most recent common ancestor of* A *in* $\mathcal{T}$ denoted $\mathrm{mrca}_{\mathcal{T}}(A)$ is the vertex v such that for every $a_i \in A$ there is a path $v \rightsquigarrow a_i$ but there is no other vertex u such that $v \rightsquigarrow u$ and for every $a_i \in A$ there is a path such that $u \rightsquigarrow a_i$.

The set of all leaves that are descended from some edge $e \in E(\mathcal{T})$ for some phylogenetic tree with leaves labelled by X is called the *edge cluster* denoted $\mathcal{C}_{\mathcal{T}}(e)$. In general when we talk about a minimal cluster C of a tree we are uninterested in the trivial cluster that occurs when $|C| = 1$, so unless otherwise specified always assume $|C| \geqslant 2$. The union of the clusters of two or more out-edges of a vertex v is called a *vertex cluster*. If A is a subset of the label set then the *minimal vertex cluster that contains* A is the union of the clusters of the children of $\mathrm{mrca}(A)$ that contain elements of A. In spite of the previous sentence $\mathcal{C}_{\mathcal{T}}(v)$ denotes the edge cluster of the in-edge of v. A *chain* is a tuple of leaf labels $(\ell_1, \ell_2, \ldots, \ell_n)$ such that $p_{\mathcal{T}}(\ell_i) = p_{\mathcal{T}}(\ell_{i+1})$ or $p(p_{\mathcal{T}}(\ell_i)) = p_{\mathcal{T}}(\ell_{i+1})$ for $i = 1, 2, \ldots, n-1$. If p is a parent of an element of a chain $A = \{a_1, a_2, \ldots, a_n\}$ then p is *internal* if it has exactly one child vertex not in A, otherwise it is called *external*. An element of A is internal/external if its parent is internal/external. For example see Figures 3.7, 3.8 and 4.5.

If A is a subset of the set of labels $\mathcal{L}(\mathcal{T})$ then $\mathcal{T}(A)$ denotes the *minimal rooted subtree of* $\mathcal{T}$ *that connects all labels corresponding to an element of* A. Further $\mathcal{T}(A)$ with all the non-root vertices of degree two suppressed is the *restriction of* $\mathcal{T}$ *to* A denoted $\mathcal{T} \mid A$. If A is a subset

of the set of labels $\mathcal{L}(\mathcal{T})$ then the *complement* $\overline{A}$ is the set $\mathcal{L}(\mathcal{T}) - A$. If A_0, A_1, ...,A_n is a collection of subsets of $\mathcal{L}(\mathcal{T})$ (usually they will be distinct) then $\mathcal{T}(A_0) \cup \mathcal{T}(A_1) \cup \cdots \cup \mathcal{T}(A_n)$ represents the collection of trees with the edge set $E(\mathcal{T}(A_0)) \cup E(\mathcal{T}(A_1)) \cup \cdots \cup E(\mathcal{T}(A_n))$ and the vertex set $V(\mathcal{T}(A_0)) \cup V(\mathcal{T}(A_1)) \cup \cdots \cup V(\mathcal{T}(A_n))$. A *pendant subtree* $\mathcal{T}'$ of a tree $\mathcal{T}$ is a tree such that $\mathcal{T}'$ is a subtree of $\mathcal{T}$ and the label set of $\mathcal{T}'$ forms a vertex cluster in $\mathcal{T}$. Not infrequently we wish to remove a pendant subtree from a phylogenetic tree $\mathcal{T}$ or replace it with a leaf. In this thesis when a subtree of labelset A is removed it is noted as $\mathcal{T} \mid \overline{A}$, and when replacing with a leaf it is noted as $\mathcal{T} \mid \overline{A - \min A}$ where A may be regarded as the smallest value in A but in the greater scheme of things may be any value within the set. The notation for these two operations employed in papers such as Bordewich et al. (2007a) has been $\mathcal{T}[-A]$ and $\mathcal{T}[A \to a]$ with $a = \min A$ rather than a leaf that does not appear in $\mathcal{L}(\mathcal{T})$, respectively.

2.3 Forests

A *forest* $\mathcal{F} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ is a collection of phylogenetic trees. A *forest* $\mathcal{F}$ *of a tree* $\mathcal{T}$ is collection of subtrees of $\mathcal{T}$ such that the trees in $\mathcal{F}$ are edge disjoint subtrees of $\mathcal{T}$. In order to ease notation in text a forest shall be written as the set of label sets of the trees it contains, however the preeceding interpretation should always be kept in mind. An *agreement forest* $\mathcal{F}$ for a finite set $\mathcal{P} = \{\mathcal{T}, \mathcal{T}', \ldots\}$ of phylogenetic trees with a common label set is a collection of trees such that $\mathcal{F}$ is a forest for every tree in $\mathcal{P}$. For example see Figures 3.2 and 3.6.

A *maximum agreement forest* $\mathcal{F}$ for $\mathcal{P}$ is an agreement forest in which the number of elements in $\mathcal{F}$ has been minimised. If $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ is a maximum agreement forest for a set of phylogenetic trees $\mathcal{P}$ then we define $m(\mathcal{P}) = k$. Instead of writing $m(\{\mathcal{T}, \mathcal{T}'\})$ the notation $m(\mathcal{T}, \mathcal{T}')$ shall be used, likewise for the other maps. In the binary case and informally, $m(\mathcal{T}, \mathcal{T}')$ corresponds to the minimum number of edges that need to be removed from $E(\mathcal{T})$ so that the resulting collection of trees are edge disjoint subtrees of $\mathcal{T}'$.

Extending certain notations from phylogenetic trees gives the *label set* of a forest $\mathcal{L}(\mathcal{F})$ as $\bigcup_{\mathcal{T}_i \in \mathcal{F}} \mathcal{L}(\mathcal{T}_i)$. Additionally the *restriction of* $\mathcal{F}$ *to a set* A written $\mathcal{F}\,|\,A$ is the set $\{\mathcal{T}_i\,|\,A : \mathcal{T}_i \in \mathcal{F}\}$. A *subforest* $\mathcal{F}'$ *of a forest* $\mathcal{F}$ is a forest such that every $\mathcal{T}_i \in \mathcal{F}'$ is a subtree of an element of $\mathcal{F}$ and further every pair of trees in $\mathcal{F}'$ are edge disjoint in $\mathcal{F}$. Note that if $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ then $\mathcal{F}$ is a subforest of $\{\mathcal{T}\}$.

Two forests $\mathcal{F}$ and $\mathcal{F}'$ are said to be *isomorphic* if for each tree $\mathcal{T}_i \in \mathcal{F}$ there is a unique

tree $\mathcal{T}_j \in \mathcal{F}'$ such that $\mathcal{T}_i \cong \mathcal{T}_j$ and vice versa. A *binary forest* is a forest in which each tree within the forest is binary. A *binary representation* $\mathcal{F}'$ of a forest $\mathcal{F}$ is a forest such that each tree $\mathcal{T}_i \in \mathcal{F}'$ is a binary representation of a unique tree in $\mathcal{F}$. If $\mathcal{T}$ is a not-necessarily binary tree and $\mathcal{F}$ is a binary forest then $\mathcal{F}$ is an *agreement forest* for $\mathcal{T}$ if there is some contraction of the edges of the trees within $\mathcal{F}$ that give edge disjoint subtrees of $\mathcal{T}$.

2.4 NP and FPT

A problem is in P, or polynomial time, if it can be solved by a deterministic Turing machine in a polynomial amount of time. Without getting into too many details a deterministic Turing machine can be considered to be a computer with a single path of computation. A problem is in NP, or non-deterministic polynomial time, if it is solvable by a non-deterministic Turing machine in polynomial time. A non-deterministic Turing machine can be thought of as a computer with an unlimited number of parallel computational paths. A good way of summarising the above is

A problem is in P if its solution is "fast" to calculate.

A problem is in NP if its solution is "fast" to verify.

Although it is clear that any problem in P is in NP it is not yet known if the reverse is true.

There are several NP problems whose running time is (most likely) exponential time in terms of only the input size but computable in time that is polynomial in terms of the input and exponential in a parameter k. Mathematically speaking a problem of size n with a parameter k is fixed-parameter tractable (FPT) if it can be computed in $\mathcal{O}(f(k)+n^c)$ where f is an arbitrary function and c is a constant independent of both n and k. The success of FPT lies in changing a running time that relies largely upon n to one that relies largely upon k and so problems in NP may run in a reasonable time regardless of the problem size. For a more in depth discussion see Downey and Fellows (1998) and Flum and Grohe (2006).

Chapter 3

Subtree Prune and Regraft

3.1 Rooted Subtree Prune and Regraft

Traditionally, an important tool to understand and model the reticulation events introduced previously has been the graph theoretic *rooted subtree prune and regraft* or rSPR distance (Maddison, 1997; Rodrigues, Sagot, and Wakabayashi, 2001; Song and Hein, 2003), which dates back at least to Hein (1990) and has been regularly recognised as a good way to represent reticulation. (Baroni et al., 2004; Maddison, 1997; Nakhleh et al., 2005; Song and Hein, 2003) Informally, this operation cuts a subtree and reattaches it to another part of the tree. The distance between two trees is quantified by the minimum number of rooted subtree prune and regraft operations that are required to transform one tree into another. In the situation where there has been only one reticulation event rooted subtree prune and regraft accurately models the situation and in general it provides a lower bound, although it may underestimate the actual number of events. (Song and Hein, 2003, 2005)

3.2 A Brief Introduction to Rooted Subtree Prune and Regraft

Let $\mathcal{T}$ be a rooted binary phylogenetic tree with leaves labelled by X. Let A be the label set of a pendant subtree in $\mathcal{T}$. Delete the edge that connects the tree rooted at $\mathrm{mrca}_{\mathcal{T}}(A)$ to its parent then add a new vertex v' and replace some edge $\{u, v\}$ with the edges $\{u, v'\}$ and $\{v', v\}$ in $\mathcal{T} \mid \overline{A}$ and to the new vertex v' attach $\mathcal{T} \mid A$. Call the resulting binary tree $\mathcal{T}'$. This operation is a

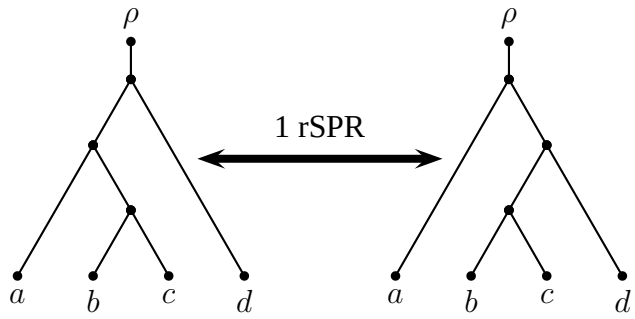


Figure 3.1: Two phylogenetic trees with a rooted subtree prune and regraft distance of 1

---

rooted subtree prune and regraft operation and we say $\mathcal{T}'$ has been obtained from $\mathcal{T}$ by a single subtree prune and regraft operation. The well defined *rooted subtree prune and regraft distance between $\mathcal{T}$ and $\mathcal{T}'$* denoted $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$ is the minimum number of rooted subtree prune and regraft operations required to transform $\mathcal{T}$ to $\mathcal{T}'$. The rooted subtree prune and regraft distance is also a biologically sound concept since reticulation events are relatively rare. (Hein, 1990)

Specifically we are interested in the following decision problem from Bordewich and Semple (2004):

PROBLEM: Binary rSPR

INSTANCE: Two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labelled by X and an integer k.

QUESTION: Is $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') \leqslant k$?

A common characterisation of the rooted subtree prune and regraft distance is via maximum agreement forests.

Theorem 3.1 (Theorem 2.1, Bordewich and Semple (2004))**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic trees with the same label set. Then*

$$d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$$

recalling that $m(\mathcal{T}, \mathcal{T}')$ is one less than the size of the agreement forest of $\mathcal{T}$ and $\mathcal{T}'$ with the fewest elements.

This characterisation of the rooted subtree prune and regraft distance via a maximum agreement forest, which dates back to Hein et al. (1996), has proven to be very fruitful. (Allen and Steel, 2001; Bordewich and Semple, 2004) Unfortunately...

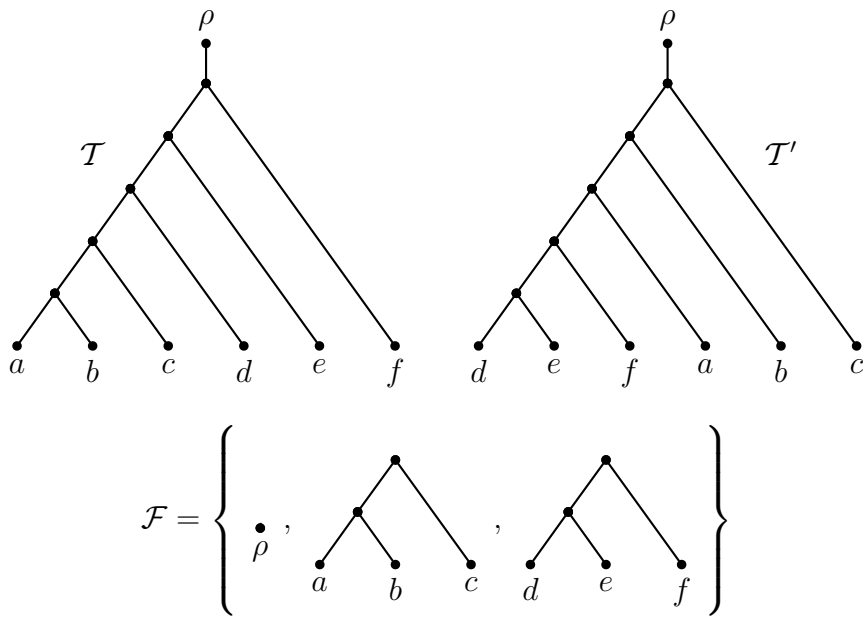


Figure 3.2: Two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ and a maximum agreement forest $\mathcal{F}$.

Theorem 3.2 (Theorem 1.1, Bordewich and Semple (2004))**.** *The decision problem* binary rSPR *is NP-complete.*

A result like this when one is trying to write algorithms to solve a problem is never a good start. Fortunately, as mentioned earlier, the problem can be kernelised, that is, it can sometimes be converted to a "smaller" problem..

Theorem 3.3 (Theorem 1.2, Bordewich and Semple (2004))**.** *The decision problem* binary rSPR *is fixed parameter tractable when parametrised by* d_{rSPR}.

To establish the fixed parameter tractability result the following reductions were used

1. Replace any pendant subtree that occurs in both trees by a single leaf with a new label as detailed in Algorithm 3.2.1.[1]

2. Replace any chain that occurs in both trees with the same orientation by three new leaves with new labels both oriented in the same direction since a chain of three leaves will re-

[1] In the code written an equivalent reduction has been used. Instead of looking for common subtrees common cherries are searched for, and—when found a—leaf in the cherry with the same label is removed from both trees.

Algorithm 3.2.1: SUBTREEEREDUCTION($\mathcal{T}, \mathcal{T}'$)

procedure SUBTREEEREDUCTION$(\mathcal{T}, \mathcal{T}')$

$A \leftarrow$ the label set of a maximal common subtree of $\mathcal{T}$ and $\mathcal{T}'$

if $|A| > 1$

then $\begin{cases} \mathcal{T} \leftarrow \mathcal{T} \mid \overline{A - \min A} \\ \mathcal{T}' \leftarrow \mathcal{T}' \mid \overline{A - \min A} \\ (\mathcal{T}, \mathcal{T}') \leftarrow \text{SUBTREEEREDUCTION}(\mathcal{T}, \mathcal{T}') \end{cases}$

return $(\mathcal{T}, \mathcal{T}')$

---

main together in some maximum agreement forest. This algorithm is given in Algorithm 3.2.2[2]

Rather than the formally stated decision problem we shall focus on the aforementioned problem of *given two phylogenetic trees what is the d_{rSPR} distance between them?* Given two binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labelled by X a naïve algorithm would try deleting all combinations of edges in one of the trees and taking the size of the agreement forest with the fewest elements, following the result of theorem 3.1, giving a running time $\mathcal{O}((2|X|)^{2k})$ to achieve this. By utilising the reductions above one obtains the following lemma.

Lemma 3.4 (Lemma 3.3, Bordewich and Semple (2004))**.** *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two rooted binary phylogenetic trees with leaves labelled by X. Let $\widehat{\mathcal{T}_1}$ and $\widehat{\mathcal{T}_2}$ be rooted binary phylogenetic trees with label set $\widehat{X}$ obtained from $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively by applying Rules 1 and 2 repeatedly until no further reduction is possible. Then $|\widehat{X}| \leqslant 28\, d_{\mathrm{rSPR}}(\mathcal{T}_1, \mathcal{T}_2)$.*

It then follows that the running time of the algorithm to find the rooted subtree prune and regraft distance is $\mathcal{O}((56k)^{2k} + |X|^3)$ where $\mathcal{O}(|X|^3)$ is the order of the running time required to do the reductions. (Bordewich and Semple, 2004)

[2]However in the code an equivalent reduction was used where any common chain was replaced with a chain of length 2 and a requirement added that the two labels appear together in the agreement forest.

Algorithm 3.2.2: rSPR-CHAINREDUCTION($\mathcal{T}, \mathcal{T}'$)

procedure rSPR-CHAINREDUCTION$(\mathcal{T}, \mathcal{T}')$
$(a_1, \ldots, a_n) \leftarrow$ maximal common chain of $\mathcal{T}$ and $\mathcal{T}'$
if $n > 3$

$$\textbf{then} \begin{cases} A \leftarrow \{a_1, \ldots, a_n\} \\ \mathcal{T} \leftarrow \mathcal{T} \mid \overline{A - \{a_1, a_2, a_n\}} \\ \mathcal{T}' \leftarrow \mathcal{T}' \mid \overline{A - \{a_1, a_2, a_n\}} \\ (\mathcal{T}, \mathcal{T}') \leftarrow \text{rSPR-CHAINREDUCTION}(\mathcal{T}, \mathcal{T}') \end{cases}$$

return $(\mathcal{T}, \mathcal{T}')$

In Bordewich et al. (2007b) an algorithm is devised that calculates the rooted subtree prune and regraft distance efficiently by reducing the edges that may need to be deleted to obtain a maximum agreement forest to as few as possible. Since the interleaving algorithm that is one of the focuses of this chapter depends on that algorithm what follows shall be a duplication of some of the biggest theorems and definitions of that paper. The first consideration will be that arising from incompatible triples.

Definition 3.5. *A* triple *of $\mathcal{T}$ is a set of three leaves $ab|c$ such that $\mathcal{T} \mid \{a, b, c\}$ is a tree with a and b a cherry, that is $p_{\mathcal{T}|\{a,b,c\}}(a) = p_{\mathcal{T}|\{a,b,c\}}(b)$. An* Incompatible Triple of $\mathcal{T}$ with respect to $\mathcal{T}'$ *is a triple such that $ab|c$ is a triple in $\mathcal{T}$ but not in $\mathcal{T}'$.*

In addition an order was defined.

Definition 3.6. *Let $ab|c$ and $xy|z$ be triples in $\mathcal{T}$, then $ab|c < xy|z$ if either*

- $\mathrm{mrca}_{\mathcal{T}}(x, y, z) \rightsquigarrow \mathrm{mrca}_{\mathcal{T}}(a, b, c)$*, or*
- $\mathrm{mrca}_{\mathcal{T}}(x, y, z) = \mathrm{mrca}_{\mathcal{T}}(a, b, c)$ *and* $\mathrm{mrca}_{\mathcal{T}}(x, y) \rightsquigarrow \mathrm{mrca}_{\mathcal{T}}(a, b)$.

A triple is minimal *if it is minimal with regards to this order.*

The following edges are also distinguished.

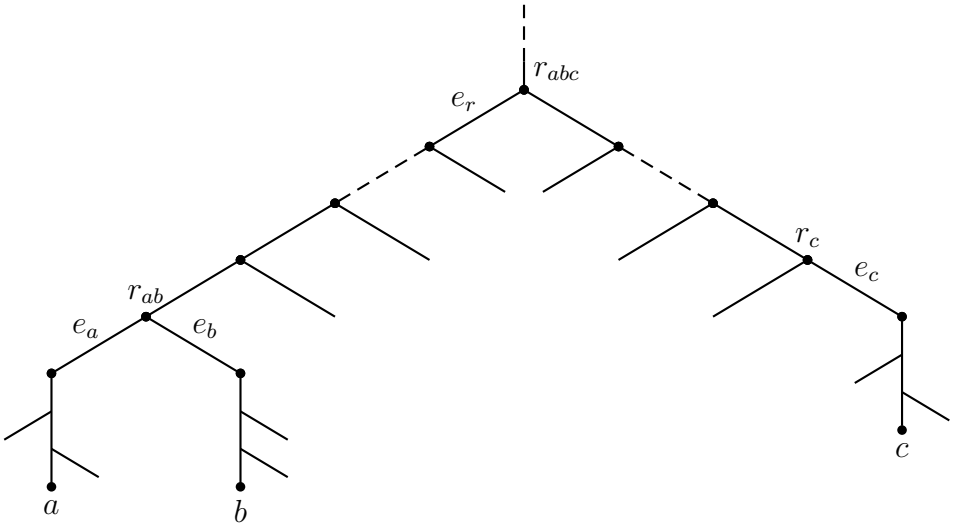


Figure 3.3: The layout of a minimal incompatible triple. A simplified version of Figure 3 in Bordewich et al. (2007b).

Definition 3.7. *For a pair of phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labelled by X such that $ab|c$ is a minimal incompatible triple of $\mathcal{T}$ with respect to $\mathcal{T}'$ let e_a be the edge adjacent to the child of $\mathrm{mrca}_{\mathcal{T}}(a,b)$ whose descendants contain a. Similarly e_b is defined for b. e_r is the edge adjacent to $\mathrm{mrca}_{\mathcal{T}}(a,b,c)$ whose descendants contain $\mathrm{mrca}_{\mathcal{T}}(a,b)$. Finally e_c is the edge on the path between $\mathrm{mrca}_{\mathcal{T}}(a,b,c)$ and c such that for all $c' \in \mathcal{C}_{\mathcal{T}}(e_c) - c$ we get $cc'|a$ and $cc'|b$ are both triples of $\mathcal{T}$ and $\mathcal{T}'$.*

These edges are illustrated in Figure 3.3. Lastly, the following lemma was proven in order to reduce the number of edges that needed to be considered in constructing a maximum agreement forest.

Lemma 3.8 (Lemma 4.1 (i), Bordewich et al. (2007b))**.** *Let $\mathcal{T}'$ be a rooted binary phylogenetic tree and $\mathcal{F}$ a rooted binary forest. If there exists a minimal incompatible triple $ab|c$ in a tree $\mathcal{T}$ of $\mathcal{F}$ with respect to $\mathcal{T}'$, then for some $i \in \{a,b,c,r\}$*

$$m(\mathcal{F} - e_i, \mathcal{T}') = m(\mathcal{F}, \mathcal{T}') - 1$$

Where $\mathcal{F} - e$ for some e in the edge set of a tree $\mathcal{T}_i$ in forest represents the forest that occurs when $\mathcal{T}_i$ is removed from the forest, and the two phylogenetic trees $\mathcal{T}_i \mid \mathcal{C}(e)$ and $\mathcal{T}_i \mid \overline{\mathcal{C}(e)}$ are added, or equivalently

$$\mathcal{F} - e = (\mathcal{F} \mid \mathcal{C}(e)) \cup (\mathcal{F} \mid \overline{\mathcal{C}(e)})$$

Additionally the concept of maximum agreement forests has been extended to forests such that if $\mathcal{F}$ is a forest and $\mathcal{T}$ is a phylogenetic tree then $m(\mathcal{F}, \mathcal{T})$ is the size of the smallest forest $\mathcal{F}'$ such that $\mathcal{F}'$ is a subforest of $\mathcal{F}$ and $\mathcal{F}'$ is a forest of $\mathcal{T}$. This also has a quite natural interpretation where it is the smallest number of rooted subtree prune and regraft moves required to change $\mathcal{F}$ into a forest for $\mathcal{T}$.

The other assistant in Bordewich et al. (2007b) involves overlapping components.

Definition 3.9. *For a forest $\mathcal{F}$ and phylogenetic tree $\mathcal{T}'$ then $\mathcal{T}_s$ and $\mathcal{T}_t$ are* overlapping components *if $\mathcal{T}'(\mathcal{L}(\mathcal{T}_s)) \cup \mathcal{T}'(\mathcal{L}(\mathcal{T}_t))$ is connected.*

Again certain edges were distinguished.

Definition 3.10. *Define $v_{st} \in V(\mathcal{T}'(\mathcal{L}(\mathcal{T}_s)) \cap \mathcal{T}'(\mathcal{L}(\mathcal{T}_t)))$ to be a minimal vertex with regards to the natural ordering obtained by considering v to be less than u if there is a directed path from v to u. Let $e_t \in E(\mathcal{T}_t)$ be such that $\mathcal{C}_{\mathcal{T}_t}(e_t) = \mathcal{C}_{\mathcal{T}'}(v_{st}) \cap \mathcal{L}(\mathcal{T}_t)$ and $e_s \in E(\mathcal{T}_s)$ be such that $\mathcal{C}_{\mathcal{T}_s}(e_s) = \mathcal{C}_{\mathcal{T}'}(v_{st}) \cap \mathcal{L}(\mathcal{T}_s)$.*

Finally, in the case that there are no incompatible triples, but there are overlapping components there are only two edges that needed to be considered for removal as pictured in Figure 3.4.

Lemma 3.11 (Lemma 4.1 (ii), Bordewich et al. (2007b))**.** *Let $\mathcal{T}'$ be a rooted binary phylogenetic tree and $\mathcal{F}$ a rooted binary forest. If there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, but there exist two components $\mathcal{T}_s$ and $\mathcal{T}_t$ of $\mathcal{F}$ that overlap in $\mathcal{T}'$, then for some $j \in \{s, t\}$*

$$m(\mathcal{F} - e_j, \mathcal{T}') = m(\mathcal{F}, \mathcal{T}') - 1$$

The running time for the bounded search algorithm rSPR-EXACT from Bordewich et al. (2007b) taking advantage of the above is $\mathcal{O}(4^k|X|^4)$ but if the reductions are included this is improved to $\mathcal{O}(4^k k^4 + |X|^3)$. (Bordewich et al., 2007b)

When compared to the problem of finding the hybridisation number discussed later in this thesis there is one other reduction not yet discussed: cluster reduction. Such a reduction for the rooted subtree prune and regraft distance ended up being somewhat more complicated and this result was published in Linz and Semple (in press). Reworded slightly their result is as follows: let A be a common cluster of two phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$. Either

- There is a maximum agreement forest $\mathcal{F}$ for $\mathcal{T} \mid (A \cup \{\rho\})$ and $\mathcal{T}' \mid (A \cup \{\rho\})$ such that $\{\rho\} \in \mathcal{F}$ in which case

$$d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T} \mid (A \cup \{\rho\}), \mathcal{T}' \mid (A \cup \{\rho\})) + d_{\mathrm{rSPR}}(\mathcal{T} \mid \overline{A}, \mathcal{T}' \mid \overline{A})$$

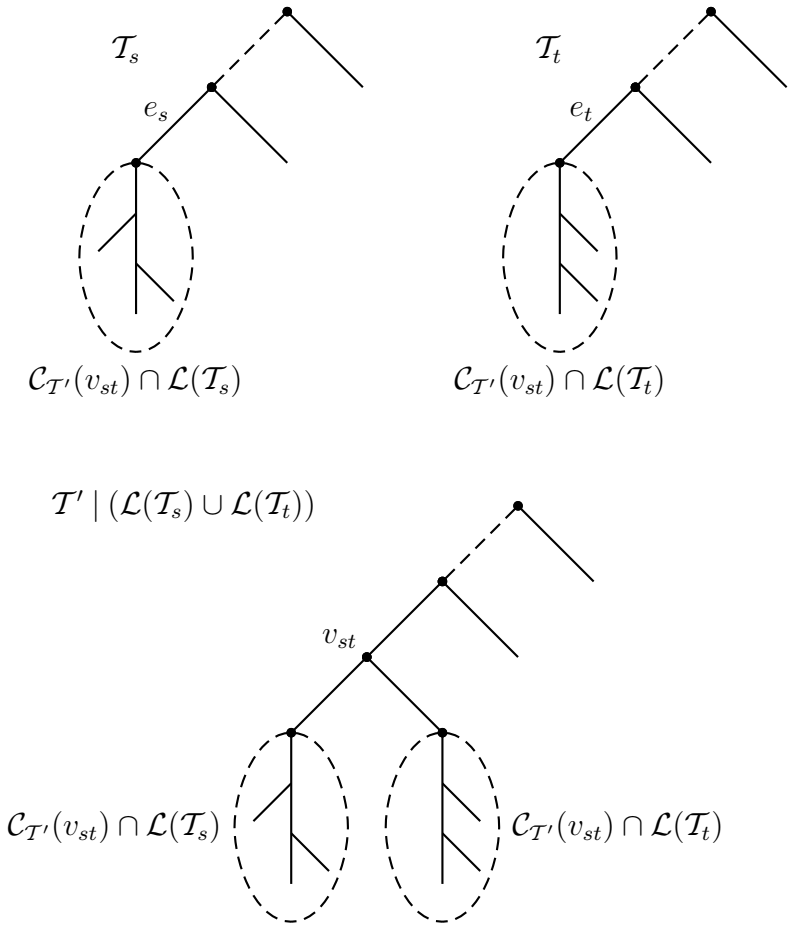


Figure 3.4: A pair of overlapping components $\mathcal{T}_s$ and $\mathcal{T}_t$. A simplified version of Figure 4 in Bordewich et al. (2007b).

or

- There is no maximum agreement forest $\mathcal{F}$ for $\mathcal{T} \mid (A \cup \{\rho\})$ and $\mathcal{T}' \mid (A \cup \{\rho\})$ such that $\{\rho\} \in \mathcal{F}$ in which case

$$d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T}|(A \cup \{\rho\}), \mathcal{T}'|(A \cup \{\rho\})) + d_{\mathrm{rSPR}}(\mathcal{T}|\overline{A - \min A}, \mathcal{T}'|\overline{A - \min A})$$

Basically, the above gives the algorithm where the agreement forests for the clusters with roots attached are initially found in the usual way. If a maximum agreement forest is found for these clusters with the root isolated then the agreement forest for the trees restricted to the remaining labels may be found since, in the maximum agreement forest under consideration, the most recent common ancestor of the cluster is an isolated, unlabeled, vertex and thus contributes nothing to the maximum agreement forest. On the other hand, the root of the cluster is connected to labels, and thus cannot be disregarded.

3.3 Non-Binary Rooted Subtree Prune and Regraft

In Linz and Semple (2008) a non-binary algorithm was devised to calculate the related problem of finding the minimum number of reticulation events for a set of trees, or hybridisation number. This is due to the fact that often reconstructed trees contain *polytomies* or *multifurcations*. Polytomies can be one of two types: *soft* in which there is an order of divergence but there is insufficient information by which to work out the order; or *hard* which is when multiple divergences have actually happened simultaneously. In practice hard polytomies are rare so it is assumed that any polytomy is soft. Thus the definition for non-binary rooted subtree prune and regraft distance follows that of the non-binary hybridisation number.

Definition 3.12. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two phylogenetic trees with the same label set. Then*

$$d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = \min_{\substack{\mathcal{B} \text{ is a binary refinement of } \mathcal{T} \\ \mathcal{B}' \text{ is a binary refinement of } \mathcal{T}'}} d_{\mathrm{rSPR}}(\mathcal{B}, \mathcal{B}')$$

First note that since the rooted subtree prune and regraft distance between two binary trees is well defined, and since every non-binary tree displays a binary tree, this is well defined. Next, that the characterisation of d_{rSPR} as a maximum agreement forest is shown to hold for non-binary trees. We begin with two lemmas.

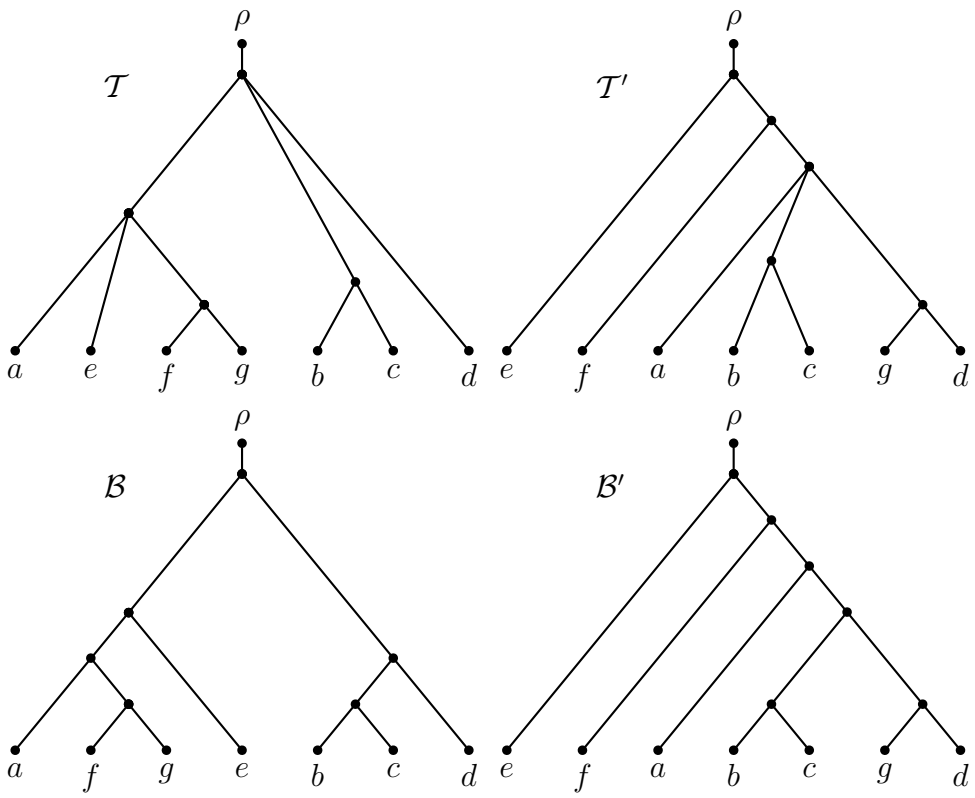


Figure 3.5: Two rooted phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ and their binary refinements $\mathcal{B}$ and $\mathcal{B}'$ as constructed by the proof in lemma 3.13.

Lemma 3.13. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted phylogenetic trees with leaves labelled by X, and let $\mathcal{F}$ be an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Then there exist binary refinements $\mathcal{B}$ and $\mathcal{B}'$ of $\mathcal{T}$ and $\mathcal{T}'$ respectively such that there is a binary refinement of $\mathcal{F}$ that is an agreement forest for $\mathcal{B}$ and $\mathcal{B}'$.*

Proof. Suppose $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \ldots, \mathcal{T}_k\}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. If $k = 0$ then the forest consists of a single tree and clearly taking any common binary refinement of $\mathcal{T}$ and $\mathcal{T}'$ will have an agreement forest of that tree, which in turn will be a binary refinement of $\mathcal{T}_\rho$.

First the result shall be shown to hold for portions of the trees that do not occur below the member of the agreement forest under consideration. Assume that the result holds for all agreement forests of size k. Let $\mathcal{F}$ be an agreement forest of size $k+1$. Take $\mathcal{T}_i \in \mathcal{F}$ for $i \neq \rho$. Let C be the minimal vertex cluster that contains $\mathcal{L}(\mathcal{T}_i)$. Clearly $\mathcal{F}_C = \mathcal{F} \mid \overline{C}$ is an agreement forest for $\mathcal{T}_C = \mathcal{T} \mid \overline{C}$ and $\mathcal{T}'_C = \mathcal{T}' \mid \overline{C}$ and by the inductive hypothesis there are binary refinements of these trees $\mathcal{B}_C$ and $\mathcal{B}'_C$ such that a binary refinement of $\mathcal{F}_C$ is also an agreement forest for $\mathcal{B}_C$ and $\mathcal{B}'_C$.

Now a binary refinement of the member of the agreement forest will be attached. Let v be the vertex of $\mathcal{T}$ such that $\mathcal{C}_\mathcal{T}(v)$ minimally properly contains $\mathcal{L}(\mathcal{T}_i)$. Then $\mathcal{C}_\mathcal{T}(v) - C$ is a cluster of $\mathcal{T}_C$ and, as $\mathcal{B}_C$ is a refinement of $\mathcal{T}_C$, it is also a cluster of $\mathcal{B}_C$. Set v_C to be the vertex in $\mathcal{B}_C$ such that $\mathcal{C}_{\mathcal{B}_C}(v_C) = \mathcal{C}_\mathcal{T}(v) - C$. Subdivide the in edge of v_C and attach a binary refinement of $\mathcal{T}_i$. Call this new tree $\mathcal{B}''$.

Now for each $\mathcal{T}_j \in \mathcal{F}$ whose label set is contained within C note that the forest restricted to each of these trees is again a proper subset of the agreement forest and thus that each has a binary refinement with the same agreement forest subset. Find the vertex v_j which is in the vertex set of $\mathcal{T}(\mathcal{L}(\mathcal{T}_i))$ and minimally contains $\mathcal{L}(\mathcal{T}_j)$. In the binary $\mathcal{B}''$ subdivide the in edge of $\mathrm{mrca}_{\mathcal{B}''}(\mathcal{C}_{\mathcal{T}|\mathcal{L}(\mathcal{T}_i)}(v_j))$ and attach a binary refinement of $\mathcal{T}_j$ such that the inductive hypothesis holds.

After the preceding paragraph has been completed by construction we have a tree $\mathcal{B}''$ which is a binary refinement of $\mathcal{T}$ and of which a binary refinement of $\mathcal{F}$ is a forest. Completing the previous steps for $\mathcal{T}'$ will give the desired result. □

The informal description of the above is as follows. If we are considering $\mathcal{T}$ and $\mathcal{T}'$ restricted to everything other than some element of a forest then that element either was pendant, in which case it was removed, or not, in which case it collapses down to a vertex. The former

case is similar to the situation in Lemma 4.2 of Linz and Semple (2008). In the latter case the above proof takes the vertex and rebuilds it into a element that is binary without changing the surrounding subtrees and thus the agreement forest.

Lemma 3.14. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted phylogenetic trees, and let $\mathcal{B}$ and $\mathcal{B}'$ be binary refinements of $\mathcal{T}$ and $\mathcal{T}'$ respectively. If $\mathcal{F}$ is an agreement forest for $\mathcal{B}$ and $\mathcal{B}'$, then $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$.*

Proof. Let $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ be an agreement forest for $\mathcal{B}$ and $\mathcal{B}'$. Since $\mathcal{B}$ and $\mathcal{B}'$ are refinements of $\mathcal{T}$ and $\mathcal{T}'$ it follows trivially that $\mathcal{B} \mid \mathcal{L}(\mathcal{T}_i)$ is a binary refinement of $\mathcal{T} \mid \mathcal{L}(\mathcal{T}_i)$ and $\mathcal{T}' \mid \mathcal{L}(\mathcal{T}_i)$. It is also clear that the trees $\mathcal{T}(\mathcal{L}(\mathcal{T}_i))$ are edge disjoint in $\mathcal{T}$ and $\mathcal{T}'$. Thus $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. □

Theorem 3.15. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted phylogenetic trees with a common label set. Then*

$$d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$$

Proof. Let $\mathcal{B}$ and $\mathcal{B}'$ be binary refinements of $\mathcal{T}$ and $\mathcal{T}'$ respectively that satisfy Lemma 3.13. Then $m(\mathcal{T}, \mathcal{T}') \geqslant m(\mathcal{B}, \mathcal{B}') = d_{\mathrm{rSPR}}(\mathcal{B}, \mathcal{B}')$ and from definition 3.12 we have $d_{\mathrm{rSPR}}(\mathcal{B}, \mathcal{B}') \geqslant d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$ so that $m(\mathcal{T}, \mathcal{T}') \geqslant d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$.

Let $\mathcal{B}$ and $\mathcal{B}'$ be binary refinements of $\mathcal{T}$ and $\mathcal{T}'$ such that $d_{\mathrm{rSPR}}(\mathcal{B}, \mathcal{B}') = d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. By Lemma 3.1 there is an agreement forest $\mathcal{F}$ such that $|\mathcal{F}| - 1 = d_{\mathrm{rSPR}}(\mathcal{B}, \mathcal{B}')$ and by Lemma 3.14 $\mathcal{F}$ is an agreement forest of $\mathcal{T}$ and $\mathcal{T}'$ so $m(\mathcal{T}, \mathcal{T}') \leqslant |\mathcal{F}| - 1 = d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. □

Now we have Theorem 3.15 under our belt clearly the next step is to show that the decision problem

PROBLEM: rSPR

INSTANCE: Two rooted phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labelled by X and an integer k.

QUESTION: Is $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') \leqslant k$.

is fixed parameter tractable. First note that since the binary case is a special case of the above it trivially follows that for arbitrary trees the problem is NP-hard. It should be clear that we may use the subtree reduction and cluster reduction as before. However, it may come as a surprise that, despite the complexity of the non-binary short and long chain reductions in the hybridisation problem (discussed later) compared to its binary counterpart, the non-binary rooted subtree

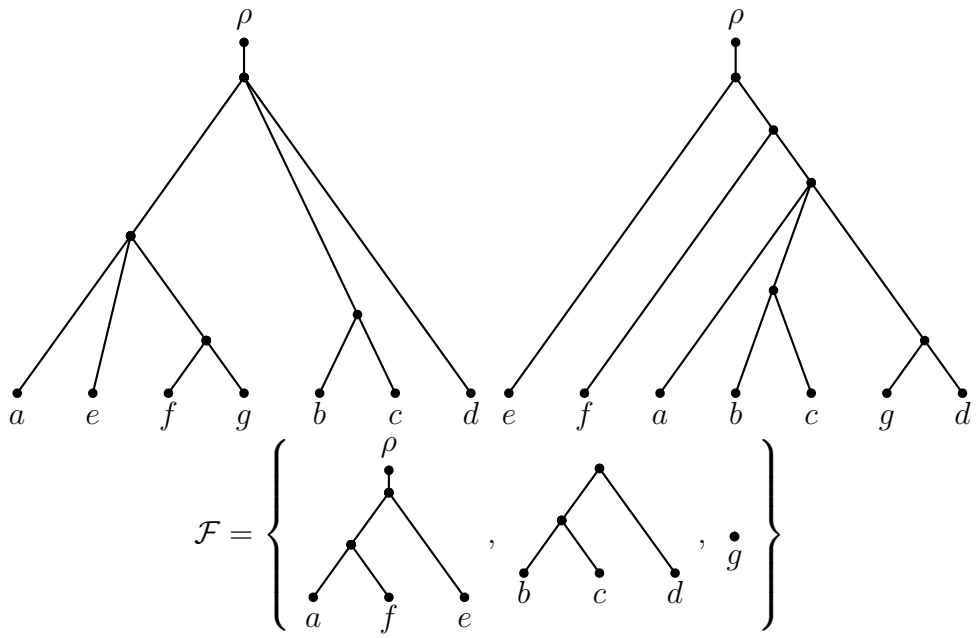


Figure 3.6: Two phylogenetic trees and a maximum agreement forest $\mathcal{F}$.

---

prune and regraft chain reduction follows from the binary case with minor modifications. A result that says if certain results are met then a set of three leaves must be in a common label set in some maximum agreement forest. In a sense there is an easier aspect to this problem as well as one that requires greater care. The easy part comes from what in Linz and Semple (2008) are called short chains, where a chain in one of the trees has in total one parent for all its leaves. However, this problem becomes more complex without care because it is possible to have two chains which share an internal edge in one of the trees, an undesirable property; for one chain to be kept together the other must be broken.

Lemma 3.16. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted phylogenetic trees with leaves labelled by X, with no common subtrees and a chain (a, b, c) such that if a and c are external and in some tree, without loss of generality say $\mathcal{T}$, we have if $p_{\mathcal{T}}(a) \neq p_{\mathcal{T}}(c)$ then the chain in $\mathcal{T}$ has at least one internal parent, then there is some maximum agreement forest in which a, b and c are contained within some common label set.*

Proof. Let $A = \{a, b, c\}$ and c be one of the lowest leaves in the chain and set $\mathcal{L}_a$ to be the labels of the leaves not occurring below the chain in $\mathcal{T}$, thus $\mathcal{L}_a = \overline{\mathcal{C}(p_{\mathcal{T}}(c))} - A$. Additionally set $\mathcal{L}_c$ to be all the labels occurring below the chain in the same tree $\mathcal{L}_c = \mathcal{C}(p_{\mathcal{T}}(c)) - A$. Similarly set $\mathcal{L}'_a$ and $\mathcal{L}'_c$ for $\mathcal{T}'$ as $\mathcal{L}'_a = \overline{\mathcal{C}(p_{\mathcal{T}'}(c))} - A$ and $\mathcal{L}'_c = \mathcal{C}(p_{\mathcal{T}'}(c)) - A$. If it exists let i be such that $\mathcal{L}_i \cap \mathcal{L}_a \neq \emptyset$ and $\mathcal{L}_i \cap \mathcal{L}_c \neq \emptyset$ and similarly j such that $\mathcal{L}_j \cap \mathcal{L}'_a \neq \emptyset$ and $\mathcal{L}_j \cap \mathcal{L}'_b \neq \emptyset$ for some

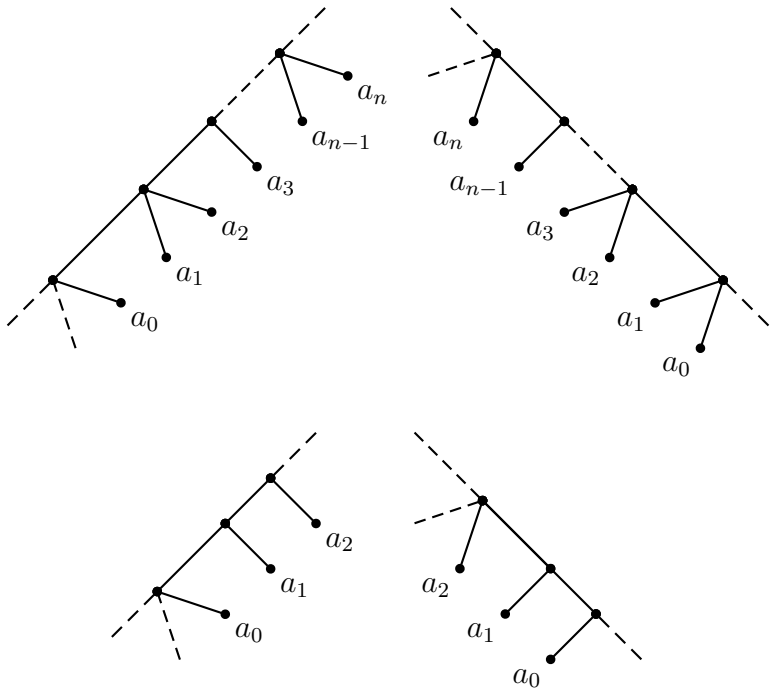


Figure 3.7: rSPR non-binary long chain reduction.

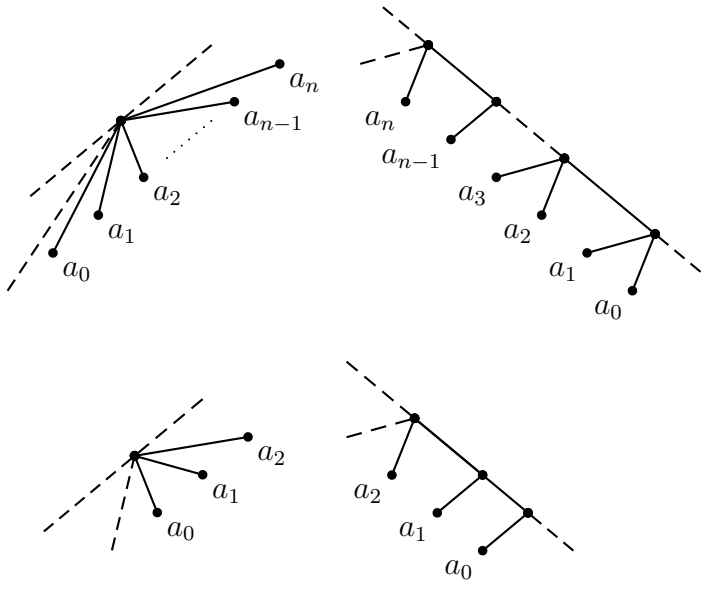


Figure 3.8: rSPR non-binary short chain reduction.

$\mathcal{L}_i$ and $\mathcal{L}_j$ label sets of trees $\mathcal{T}_i$ and $\mathcal{T}_j$ in a maximum agreement forest for $\mathcal{T}$ and $\mathcal{T}'$.

There are six cases:

1. There is no such i or j;
2. there is an i but no j;
3. there is a j but no i;
4. i and j exist with $i \neq j$;
5. i and j exist with $i = j$ and $\mathcal{L}_i \cap \mathcal{L}_c \cap \mathcal{L}'_c = \emptyset$;
6. i and j exist with $i = j$ and $\mathcal{L}_i \cap \mathcal{L}_c \cap \mathcal{L}'_c \neq \emptyset$.

In case 1 let $\mathcal{L}_b$ be the label set that contains b. If $b = \mathcal{L}_b$ then an agreement forest of the same size may be obtained by detaching a and c from their respective trees and creating a new tree $\mathcal{T} \mid \{a, b, c\}$. Otherwise let $\mathcal{L}_a$ be the label set containing a and $\mathcal{L}_c$ be the label set containing c. An equal or smaller agreement forest may be obtained by replacing the not necessarily distinct $\mathcal{T} \mid \mathcal{L}_a$, $\mathcal{T} \mid \mathcal{L}_b$ and $\mathcal{T} \mid \mathcal{L}_c$ with $\mathcal{T} \mid (\mathcal{L}_a \cup \mathcal{L}_b \cup \mathcal{L}_c)$.

Since cases 2 and 3 are equivalent consider only 2. Unless in $\mathcal{T}$ we have $p_{\mathcal{T}}(a) = p_{\mathcal{T}}(b) = p_{\mathcal{T}}(c)$ then some subset of A must be a label set in the forest, noting that if it is not proper then we already have the required result. If $p_{\mathcal{T}}(a) = p_{\mathcal{T}}(c)$ then replace not necessarily distinct $\mathcal{T} \mid \mathcal{L}_a$, $\mathcal{T} \mid \mathcal{L}_b$ and $\mathcal{T} \mid \mathcal{L}_c$ with $\mathcal{T} \mid (\mathcal{L}_a \cup \mathcal{L}_b \cup \mathcal{L}_c)$. Otherwise replace $\mathcal{T}_i$ and not necessarily distinct $\mathcal{T} \mid \mathcal{L}_a$, $\mathcal{T} \mid \mathcal{L}_b$ and $\mathcal{T} \mid \mathcal{L}_c$ with $\mathcal{T}_i \mid (\mathcal{C}(p_{\mathcal{T}}(c)) - A)$, $\mathcal{T}_i \mid (\overline{\mathcal{C}(p_{\mathcal{T}}(c))} - A)$ and $\mathcal{T} \mid (\mathcal{L}_a \cup \mathcal{L}_b \cup \mathcal{L}_c)$ noting that at least two of $\mathcal{L}_a$, $\mathcal{L}_b$ and $\mathcal{L}_c$ must be unequal.

For case 4 if $p(a) = p(c)$ in one of the trees then it effectively reduces to the above paragraph so assume this is not the case. Since we have no common subtrees we have that none of $\mathcal{L}_a$, $\mathcal{L}_b$ and $\mathcal{L}_c$ are equal. Thus replace $\mathcal{T}_i$, $\mathcal{T}_j$, $\mathcal{T} \mid \mathcal{L}_a$, $\mathcal{T} \mid \mathcal{L}_b$ and $\mathcal{T} \mid \mathcal{L}_c$ with $\mathcal{T}_i \mid (\mathcal{C}(p_{\mathcal{T}}(c)) - A)$, $\mathcal{T}_i \mid (\overline{\mathcal{C}(p_{\mathcal{T}}(c))} - A)$, $\mathcal{T}_j \mid (\mathcal{C}(p_{\mathcal{T}'}(c)) - A)$, $\mathcal{T}_j \mid (\overline{\mathcal{C}(p_{\mathcal{T}'}(c))} - A)$ and $\mathcal{T} \mid (\mathcal{L}_a \cup \mathcal{L}_b \cup \mathcal{L}_c)$.

For case 5 again assume that $p(a) \neq p(c)$ in both trees and replace $\mathcal{T}_i$, $\mathcal{T} \mid \mathcal{L}_a$, $\mathcal{T} \mid \mathcal{L}_b$ and $\mathcal{T} \mid \mathcal{L}_c$ with $\mathcal{T}_i \mid (\mathcal{C}(p_{\mathcal{T}}(a)) - A)$, $\mathcal{T}_i \mid (\mathcal{C}(p_{\mathcal{T}'}(a)) - A)$, $\mathcal{T}_i \mid (\overline{\mathcal{C}(p_{\mathcal{T}}(a)) \cup \mathcal{C}(p_{\mathcal{T}'}(a))} - A)$ and $\mathcal{T} \mid (\mathcal{L}_a \cup \mathcal{L}_b \cup \mathcal{L}_c)$.

Finally for case 6 again $p(a) \neq p(c)$ in both trees and replace $\mathcal{T}_i$, $\mathcal{T} \mid \mathcal{L}_a$, $\mathcal{T} \mid \mathcal{L}_b$ and $\mathcal{T} \mid \mathcal{L}_c$ with $\mathcal{T}_i \mid ((\mathcal{C}(p_{\mathcal{T}}(a)) \cap \mathcal{C}(p_{\mathcal{T}'}(a))) - A)$, $\mathcal{T}_i \mid ((\mathcal{C}(p_{\mathcal{T}}(a)) \,\Delta\, \mathcal{C}(p_{\mathcal{T}'}(a))) - A)$, $\mathcal{T}_i \mid (((X - \mathcal{C}(p_{\mathcal{T}}(a))) - \mathcal{C}(p_{\mathcal{T}'}(a))) - A)$ and $\mathcal{T} \mid (\mathcal{L}_a \cup \mathcal{L}_b \cup \mathcal{L}_c)$ where $A \,\Delta\, B$ denotes the symmetric difference $(A \cup B) - (A \cap B)$. □

Using this result any chain of more than three leaves with an internal element in both trees may be replaced by a chain with three leaves. This plus that we may also utilise subtree reductions means that the problem is fixed parameter tractable.

Theorem 3.17. *The decision problem* rSPR *is fixed parameter tractable when parametrised by* d_{rSPR}.

Proof. Following Lemma 6.2 in Linz and Semple (2008) it then follows that the longest chain after reductions has length of, at most, 12. It then follows from Lemma 3.3 in Bordewich and Semple (2007b) that $|\widehat{X}| < 64\, d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. □

3.4 Rekernelisation

Finally the first rekernelisation result. Rekernelisation, also called *interleaving*, is a method that may sometimes be applied to fixed parameter algorithms. Rekernelising refers to repeated kernelisation steps while one processes the reduced data. (Niedermeier and Rossmanith, 2000) For this particular technique it seems it would be possible to construct pathological examples in which the result that follows would receive limited use, if any. As such no attempt shall be made to calculate the running time taking advantage of such a method. For example see Figure 3.9.

Definition 3.18. *Let $\mathcal{F}$ be a forest of $\mathcal{T}$. Then the* non-crossing partition of $\mathcal{F}$ with respect to $\mathcal{T}$*, written $\mathcal{P}_{\mathcal{T}}(\mathcal{F})$, is a collection of subsets of $\mathcal{F}$ such that for each $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n\} \in \mathcal{P}_{\mathcal{T}}(\mathcal{F})$ we have that $\mathcal{T}(\mathcal{L}(\mathcal{T}_0)) \cup \mathcal{T}(\mathcal{L}(\mathcal{T}_1)) \cup \cdots \cup \mathcal{T}(\mathcal{L}(\mathcal{T}_n))$ is maximally connected.*

An illustration of a non-crossing partition of a forest with respect to a tree appears in Figure 3.10.

Lemma 3.19. *Let $\mathcal{T}'$ be a phylogenetic tree and $\mathcal{F}$ a forest such that $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{T}')$.*

$$m(\mathcal{F}, \mathcal{T}') = \sum_{\mathcal{F}_i \in \mathcal{P}_{\mathcal{T}'}(\mathcal{F})} m(\mathcal{F}_i, \mathcal{T}' \mid \mathcal{L}(\mathcal{F}_i))$$

Proof. From the definition of a non-crossing partition for any $\mathcal{F}_i \in \mathcal{P}_{\mathcal{T}'}(\mathcal{F})$ we have $\mathcal{F} \mid \mathcal{L}(\mathcal{F}_i)$ is a subset of $\mathcal{F}$ thus we know that the maximum agreement forest for $\mathcal{F}$ and $\mathcal{T}'$ restricted to $\mathcal{L}(\mathcal{F}_i)$ will be an agreement forest for $\mathcal{F}_i$ and $\mathcal{T} \,|\, \mathcal{L}(\mathcal{F}_i)$. Now assume that a maximum agreement forest for $\mathcal{F}_i$ and $\mathcal{T}' \mid \mathcal{L}(\mathcal{F}_i)$ is not equal in size to a maximum agreement forest $\mathcal{F}'$ for $\mathcal{F}$ and $\mathcal{T}'$

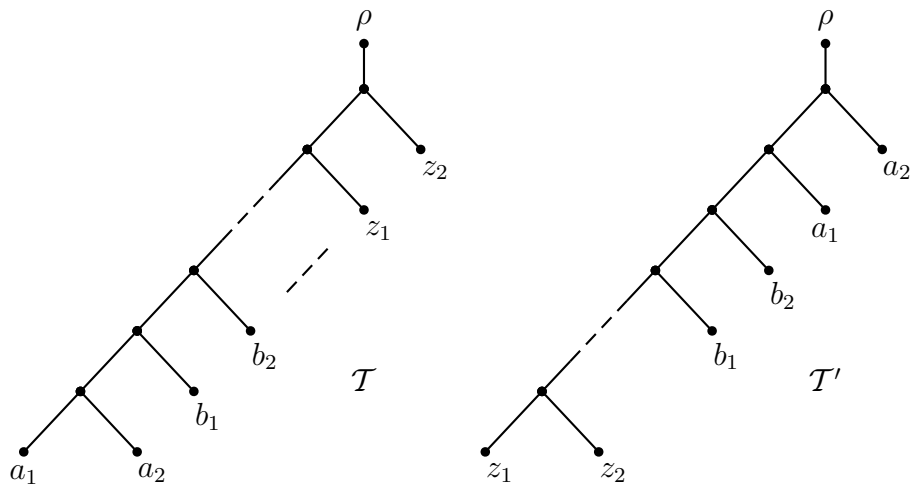


Figure 3.9: Two phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ for which rekernelisation would have little appreciable affect. Note that the label sets of the maximum agreement forest would be $\{\{\rho\}, \{a_1, a_2\}, \{b_1, b_2\}, \ldots, \{z_1, z_2\}\}$.

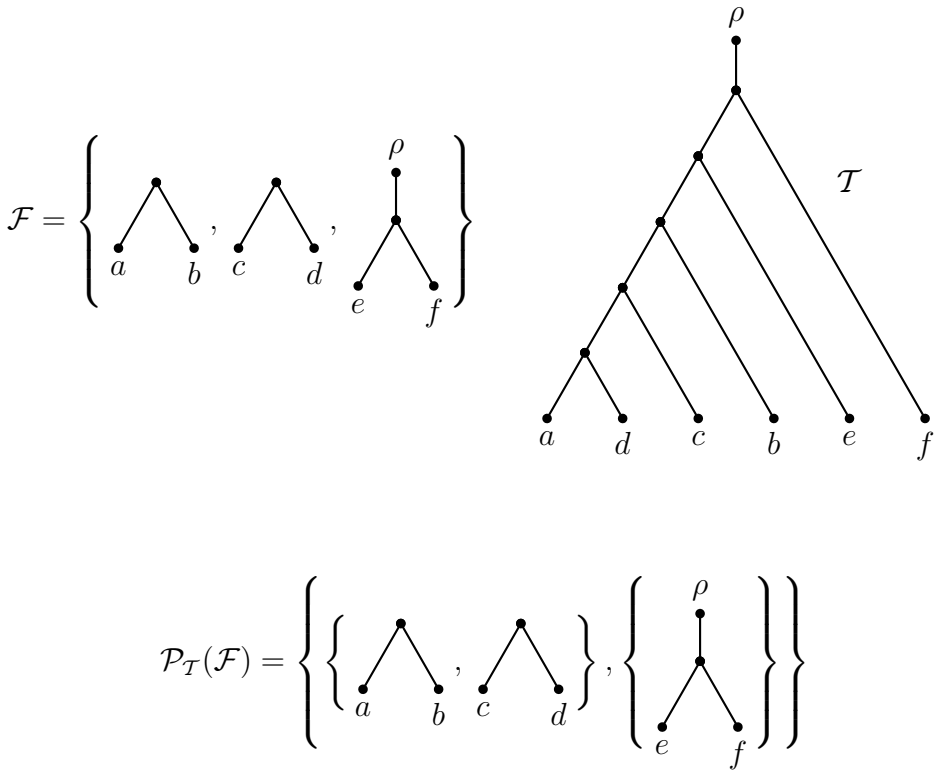


Figure 3.10: A forest $\mathcal{F}$ and tree $\mathcal{T}$ and the noncrossing partition of $\mathcal{F}$ with respect to $\mathcal{T}$.

restricted to elements with labels from $\mathcal{L}(\mathcal{F}_i)$. By definition $\mathcal{F}_i$ is a subset of $\mathcal{F}$ and further for an agreement forest $\mathcal{F}'$ for $\mathcal{F}$ and $\mathcal{T}'$ we will have $\mathcal{F}' \mid \mathcal{L}(\mathcal{F}_i)$ is a subforest of $\mathcal{F} \mid \mathcal{L}(\mathcal{F}_i)$ and a subset of $\mathcal{F}'$. The latter result follows since $\mathcal{F}'$ is a subforest of $\mathcal{F}$. It then follows that the size of $\mathcal{F}' \mid \mathcal{L}(\mathcal{F}_i)$ is equal to the size of a maximum agreement forest for $\mathcal{F}_i$ and $\mathcal{T}' \mid \mathcal{F}_i$ since any other option contradicts the maximality of one of the options.

Applying the above proof to each member of the non-crossing partition completes the proof. □

It is worth at this point noting a few things about the above result. First is that not every agreement forest obtained by the bounded search algorithm rSPR-EXACT from Bordewich et al. (2007b) may be the same as the union of the agreement forests obtained by the rekernelisation approach. The reason for this is that the reductions—including chain reduction—may be reapplied. This might involve asserting that some set of labels in the agreement forest remain together that may be split in the agreement forest obtained by bounded search. The second note is that in calculating the agreement forests for the right hand side, only one will contain the root as part of the label set. Although this may seem contradictory at first it actually ties in nicely with a number of other results, most notably the comment made in Allen and Steel (2001) where is pointed out that the root needs to be considered part of the label set to obtain the correct result when the root is isolated in the agreement forest. In light of the previous result this amounts to removing the root from the tree and then splitting based on the non-crossing partition. Also compare the cluster reduction from Linz and Semple (in press).

The results regarding reductions are always stated as being applied prior to calculating the rooted subtree prune and regraft distance, especially in this context since it is stated as in *it does not change the distance*. What is required for the rekernelising above to really have some sort of tangible affect is to determine that the reductions do not change the distance *during the run of the algorithm*. That is, rather than considering reductions on $\mathcal{T}$ and $\mathcal{T}'$, we wish to consider those on $\mathcal{F}$ and $\mathcal{T}'$.

The following results are dealing with binary trees since there is not yet an algorithm for the non-binary rooted subtree prune and regraft distance. However should a non-binary algorithm similar to rSPR-EXACT be developed it would be surprising if a similar result did not hold.

Lemma 3.20. *Let $\mathcal{F}$ be a binary forest and $\mathcal{T}'$ a binary phylogenetic tree, with $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{T}')$. Let A be the leaf set of a pendant subtree common to $\mathcal{T}'$ and $\mathcal{T}_i$ for some $\mathcal{T}_i \in \mathcal{F}$. Then any*

maximum agreement forest for $\mathcal{F}$ and $\mathcal{T}'$ contains a tree whose label set contains every element of A.

Proof. Follows trivially from Bordewich and Semple (2007b) Lemma 3.1 (a). □

Lemma 3.21. *Let $\mathcal{F}$ be a binary forest and $\mathcal{T}'$ a binary phylogenetic tree with $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{T}')$. Let $A = (a_0, a_1, \ldots, a_n)$ be a common chain with $|A| \geqslant 3$. Then there exists a maximum agreement forest $\mathcal{F}'$ such that there is a tree $\mathcal{T}_i \in \mathcal{F}'$ with $A \subseteq \mathcal{L}(\mathcal{T}_i)$.*

Proof. Follows trivially from Bordewich and Semple (2007a) Lemma 3.1. □

Lemma 3.22. *Let $\mathcal{F}$ be a binary forest and $\mathcal{T}'$ a binary phylogenetic tree and let $\widehat{\mathcal{F}}$ and $\widehat{\mathcal{T}'}$ be the respective objects after subtree and rSPR chain reduction have been applied as much as possible. Then $m(\mathcal{F}, \mathcal{T}') = m(\widehat{\mathcal{F}}, \widehat{\mathcal{T}'})$.*

Proof. This proof follows easily from the previous lemmas. □

In addition to the above the cluster reduction detailed in the introductory section may also be applied to a member of the forest $\mathcal{T}_i \in \mathcal{F}$ and $\mathcal{T}'$ with the same considerations being taken into account as to the condition of the ρ in the maximum agreement forests.

Other than the bonus of rekernelisation there are at least two comments worth making about the pseudo-code in Algorithm 3.4.1. First is: why bother with non-crossing partitions since it is essentially overlapping components? The answer to this is simple: it is easy to determine (in the binary case) when trees overlap by considering that if they do then there is a walk that passes through from the leaf to the root of one member of the forest which passes through the root of the other member of the forest in the tree in which we are considering if the components overlap. It is not so easy to determine the edges at which we need break after that. Secondly, if the edge that is removed is adjacent to a label a slight speed up can be obtained by removing that label from $\mathcal{T}'$ and $\mathcal{F}$ immediately.

3.5 Results

In Table 3.1 the bounded search algorithm rSPR-EXACT from Bordewich et al. (2007b) and the rekernelisation algorithm rSPR-REKERNELISE are applied to the grass data set from Grass Phylogeny Working Group (2001) and compared. It can be seen that rekernelising is indeed

Algorithm 3.4.1: rSPR-REKERNELISE($\mathcal{T}, \mathcal{T}'$)

procedure rSPR-REDUCE($\mathcal{F}, \mathcal{T}', k$)
 for each $\mathcal{T}_i \in \mathcal{F}$
 do $\begin{cases} (\mathcal{T}_i, \mathcal{T}') \leftarrow \text{SUBTREEEREDUCTION}(\mathcal{T}_i, \mathcal{T}') \\ (\mathcal{T}_i, \mathcal{T}') \leftarrow \text{rSPR-CHAINREDUCTION}(\mathcal{T}_i, \mathcal{T}') \\ C \leftarrow \text{labels of a minimal common cluster of } \mathcal{T}_i \text{ and } \mathcal{T}' \\ \textbf{if } 1 < |C| < |\mathcal{L}(\mathcal{T}')| \\ \quad \textbf{then} \begin{cases} k' \leftarrow \text{rSPR-EXACT}(\mathcal{T}_i \mid (C \cup \{\rho\}), \mathcal{T}' \mid (C \cup \{\rho\}), k) \\ \textbf{if } \{\rho\} \in \mathcal{F} \textbf{ for some maximum agreement forest } \mathcal{F} \textbf{ for} \\ \qquad \mathcal{T}_i \mid (C \cup \{\rho\}) \textbf{ and } \mathcal{T}' \mid (C \cup \{\rho\}) \\ \textbf{then } k' \leftarrow k' + \text{rSPR-REDUCE}(\mathcal{F} \mid \overline{C}, \mathcal{T}' \mid \overline{C}, k - k') \\ \textbf{else } k' \leftarrow k' + \text{rSPR-REDUCE}(\mathcal{F} \mid \overline{C - \min C}, \\ \qquad\qquad \mathcal{T}' \mid \overline{C - \min C}, k - k') \\ \textbf{return } (k') \end{cases} \end{cases}$
 return (rSPR-EXACT($\mathcal{F}, \mathcal{T}', k$))

procedure rSPR-EXACT($\mathcal{F}, \mathcal{T}', k$)
 if $k \leqslant 0$
 then return (0)
 else if there exists a minimal incompatible triple $ab|c$ **of** $\mathcal{F}$ **with respect to** $\mathcal{T}'$
 then $\begin{cases} k \leftarrow \min\{k, \text{rSPR-UNCROSS}(\mathcal{F} - e_a, \mathcal{T}', k-1) + 1\} \\ k \leftarrow \min\{k, \text{rSPR-UNCROSS}(\mathcal{F} - e_b, \mathcal{T}', k-1) + 1\} \\ k \leftarrow \min\{k, \text{rSPR-UNCROSS}(\mathcal{F} - e_c, \mathcal{T}', k-1) + 1\} \\ k \leftarrow \min\{k, \text{rSPR-UNCROSS}(\mathcal{F} - e_r, \mathcal{T}', k-1) + 1\} \\ \textbf{return } (k) \end{cases}$
 else if there exists a pair of components $\mathcal{T}_s$**,** $\mathcal{T}_t$ **of** $\mathcal{F}$ **that overlap in** $\mathcal{T}'$
 then $\begin{cases} k \leftarrow \min\{k, \text{rSPR-UNCROSS}(\mathcal{F} - e_s, \mathcal{T}', k-1) + 1\} \\ k \leftarrow \min\{k, \text{rSPR-UNCROSS}(\mathcal{F} - e_t, \mathcal{T}', k-1) + 1\} \\ \textbf{return } (k) \end{cases}$
 else return (0)

procedure rSPR-UNCROSS($\mathcal{F}, \mathcal{T}', k$)
 $k' \leftarrow 0$
 for each $\mathcal{F}_i \in \mathcal{P}_{\mathcal{T}'}(\mathcal{F})$
 do $\begin{cases} k' \leftarrow k' + \text{rSPR-REDUCE}(\mathcal{F}_i, \mathcal{T}' \mid \mathcal{L}(\mathcal{F}_i), k - k') \\ \textbf{if } k' \geqslant k \\ \quad \textbf{then break} \end{cases}$
 return (k')

main
 return (rSPR-REDUCE($\mathcal{T}, \mathcal{T}', |\mathcal{L}(\mathcal{T})|$))

Grasses	d_{rSPR}	Bounded	Rekernelise
ndhF/phyt	12	3.7s	2.9s
ndhF/rbcL	10	3.3s	3.4s
ndhF/rpoC2	11	14.6s	9.8s
ndhF/waxy	7	0.4s	0.4s
ndhF/ITS	19	17.6m	10.8m
phyt/rbcL	4	0.0s	0.1s
phyt/rpoC2	6	0.6s	0.8s
phyt/waxy	3	0.0s	0.0s
phyt/ITS	8	0.2s	0.3s
rbcL/rpoC2	11	43.5s	28.1s
rbcL/waxy	6	1.0s	1.1s
rbcL/ITS	13	18.4m	12.3m
rpoC2/waxy	1	0.0s	0.0s
rpoC2/ITS	14	12.6m	7.1m
waxy/ITS	7	1.0s	1.0s

Table 3.1: Time taking for the implementation of bounded search algorithm rSPR-EXACT from Bordewich et al. (2007b) and rekernelising for the binary rooted subtree prune and regraft distance on the *poaceae* data set from Grass Phylogeny Working Group (2001)

faster although not significantly. This could be due to the way that the edges removed according to the current results are always close to the leaves. It is possible that all this method is doing is taking leaves off $\mathcal{T}'$ during the run and not being used to full advantage. What other opportunities then might one garner from the method? Let us say that threaded code was written in order to take advantage of the increasing prevalence of multiple CPU computers. At present one could start a new thread upon breaking incompatible triples, or upon breaking overlapping components. However, we cannot calculate clusters and the cluster reduced tree independently since the value of the reduced tree depends upon whether or not the root is isolated in some agreement forest for the cluster. It seems like multithreading based upon the different pieces given by the non-crossing partition as well as the above paths may be a worthwhile avenue to consider when deciding when to create new threads.

Chapter 4

Phylogenetic Networks

The rooted subtree prune and regraft distance provides a lower bound for the number of reticulation events. However it also contains a fairly major unstated assumption, namely that if we are considering a tree $\mathcal{T}_i$ in an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ then the most recent common ancestor for the set of species in $\mathcal{T}_i$, namely $\mathcal{L}(\mathcal{T}_i)$ is the same hypothetical ancestor represented by the most recent common ancestor of those species in the other tree. The problem that then follows from this is that it permits species to be their own descendants. An illustration of this is in Figure 4.2. If the ability of ancestral species to travel through time is forbidden, then the concept of a phylogenetic network is born. The smallest number of reticulation events in such a setting is the subject of a number of papers including Baroni, Grünewald, Moulton, and Semple (2005); Baroni, Semple, and Steel (2006); Bordewich and Semple (2007a); Hallett and Lagergren (2004); Maddison (1997) and Nakhleh et al. (2005). Additionally, variants are considered in Gusfield and Bansal (2005); Gusfield et al. (2004); Hein (1990) and Song and Hein (2003).

4.1 A Brief Introduction to Phylogenetic Networks

A *phylogenetic network* $\mathcal{H}$ *on* X is a rooted acyclic digraph such that X, known as the *label set of* $\mathcal{H}$, labels the vertices of out-degree zero, and all vertices with out-degree one have in-degree of at least two. Vertices in $\mathcal{H}$ with in-degree at least two are called *hybridisation vertices*. A phylogenetic network is said to *display* a rooted phylogenetic tree $\mathcal{T}$ if $\mathcal{T}$ can be obtained by deleting edges and vertices and contracting the edges of $\mathcal{H}$. The set of vertices with out-degree zero are the *leaves* of a phylogenetic network.

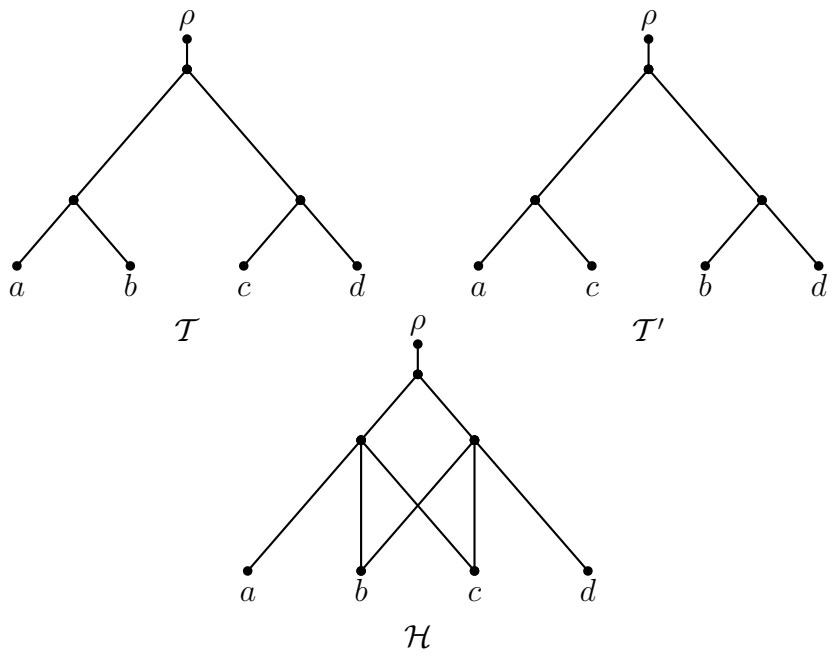


Figure 4.1: Two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ and a phylogenetic network $\mathcal{H}$ that displays them both.

The *hybridisation number* $h(\mathcal{H})$ of a phylogenetic network $\mathcal{H}$ is given by

$$h(\mathcal{H}) = \sum_{v \in V(\mathcal{H}) - \{\rho\}} d^-(v) - 1$$

Informally it is the number of times two distinct hypothetical ancestors have produced offspring. Since the data is usually presented in the form of a set $\mathcal{P}$ of rooted phylogenetic trees $\mathcal{H}$ *displays* $\mathcal{P}$ if each tree in $\mathcal{P}$ is displayed by $\mathcal{H}$. That is all the evolutionary relationships expressed in the phylogenetic trees in $\mathcal{P}$ are also expressed in $\mathcal{H}$. Extending the definition and motivation of hybridisation number to a set of phylogenetic trees $\mathcal{P}$ we obtain

$$h(\mathcal{P}) = \min_{\mathcal{H} \text{ displays } \mathcal{P}} h(\mathcal{H})$$

It turns out that the hybridisation number of a set of trees can be found via a characterisation as an agreement forest with an additional property. If $\mathcal{F}$ is an agreement forest for some set of phylogenetic trees $\mathcal{P}$ then let $G_{\mathcal{F}}$ be the digraph whose vertex set is $\mathcal{F}$ and arc set is given by

$$A(G_{\mathcal{F}}) = \{(\mathcal{T}_i, \mathcal{T}_j) : \mathrm{mrca}_{\mathcal{T}}\, \mathcal{L}(\mathcal{T}_i) \rightsquigarrow \mathrm{mrca}_{\mathcal{T}}\, \mathcal{L}(\mathcal{T}_j) \text{ for some } \mathcal{T} \in \mathcal{P}\}$$

If $G_{\mathcal{F}}$ contains a directed cycle then $\mathcal{F}$ is said to be *cyclic* otherwise $\mathcal{F}$ is *acyclic*. If a maximum

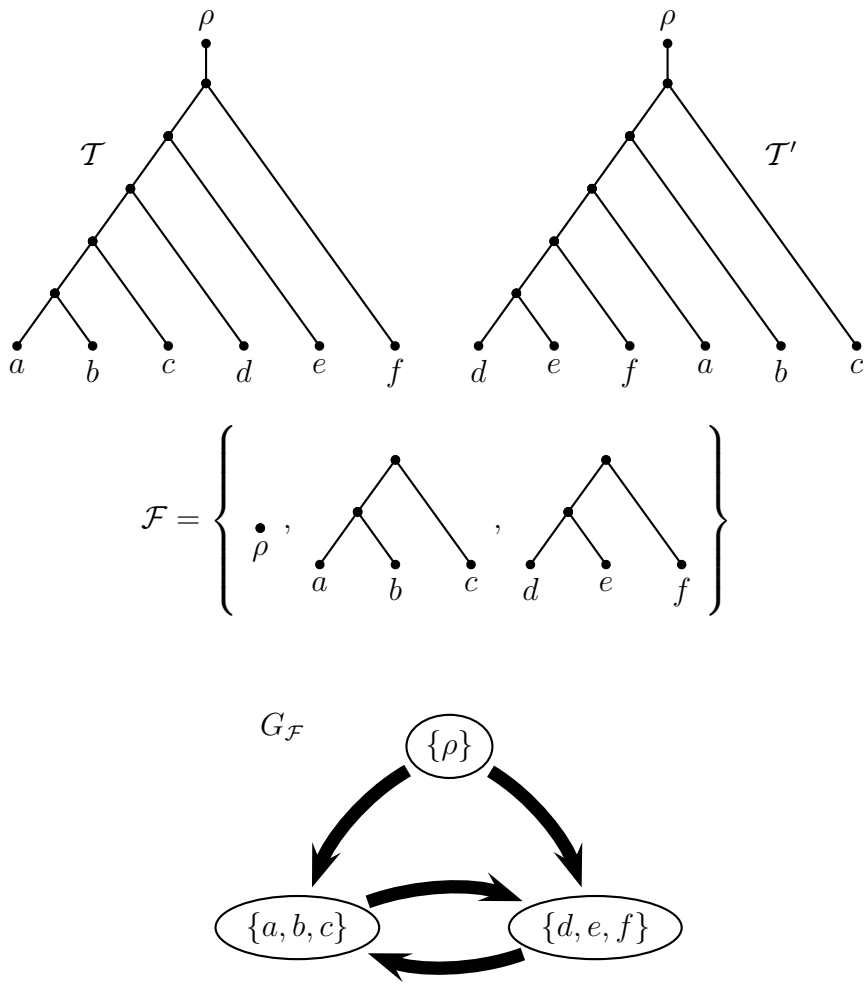


Figure 4.2: A pair of phylogenetic trees with a cyclic agreement forest.

---

agreement forest $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \ldots, \mathcal{T}_k\}$ for a set of phylogenetic trees $\mathcal{P}$ is additionally required to be acyclic then we have a *maximum acyclic agreement forest* and define $m_a(\mathcal{P}) = k$.

Theorem 4.1 (Theorem 2, Baroni et al. (2005))**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic trees. Then*

$$h(\mathcal{T}, \mathcal{T}') = m_a(\mathcal{T}, \mathcal{T}')$$

However, the hybridisation number, unlike the rooted subtree prune and regraft distance, may be calculated for any finite set of phylogenetic trees in which case the above theorem no longer holds. Thus for $\mathcal{P}$ of arbitrary size we define

$$h'(\mathcal{P}) = m_a(\mathcal{P})$$

That is the $h'(\mathcal{P})$ corresponds to the minimum number of vertices with more than one parent in

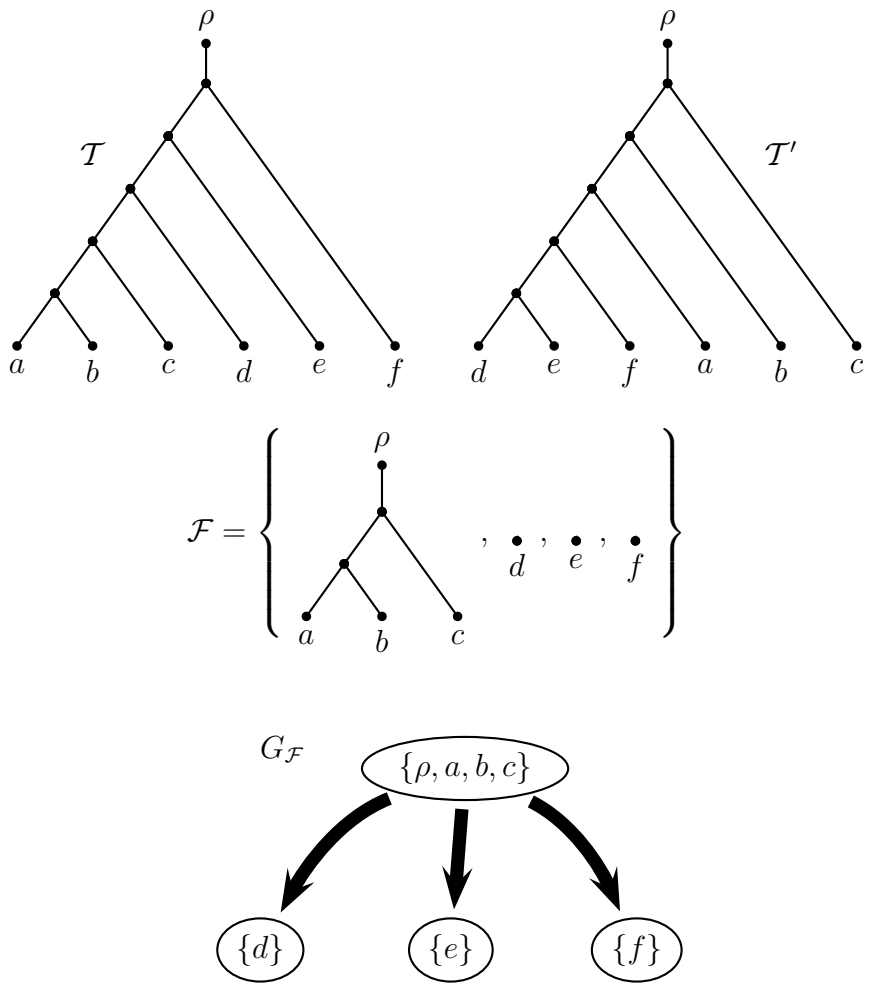


Figure 4.3: A pair of phylogenetic trees and an acyclic agreement forest.

---

any phylogenetic network that displays $\mathcal{P}$. Compare this to $h(\mathcal{P})$ which determines the smallest number of edges that must be added to obtain a phylogenetic network that displays $\mathcal{P}$.

Given that constructing an agreement forest for a pair of phylogenetic trees is NP-hard it should come as no surprise that constructing an acyclic agreement forest for a finite set of phylogenetic trees is as well.

Theorem 4.2 (Theorem 2.1, Bordewich and Semple (2007a)[1])**.** *Computing the hybridisation number between an arbitrary pair of rooted binary phylogenetic trees with the same label set is NP-hard.*

Again like the rooted subtree prune and regraft distance the problem is known to be fixed

[1]In Bordewich and Semple (2007a) it is actually shown that the problem is APX-hard which is a subclass of NP-hard that allows polynomial time approximation algorithms.

parameter tractable. In Bordewich and Semple (2007b) the problem is stated as an optimisation problem however to be consistent it shall here be stated as a decision problem.

PROBLEM: Binary Hybridisation

INSTANCE: Two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labelled by X and an integer k.

QUESTION: Is $h(\mathcal{T}, \mathcal{T}') \leqslant k$?

Theorem 4.3 (Theorem 1.1, Bordewich and Semple (2007b))**.** *The decision problem Binary Hybridisation is fixed parameter tractable with h as the parameter.*

For hybridisation it is not sufficient to simply replace a chain with three elements due to the acyclic condition. Let P be a disjoint collection of subsets of X. A set of phylogenetic trees with an associated set P and leaves labelled by X are a set of *weighted phylogenetic trees*. The weighting function w maps, in the binary case, $P \to \mathbb{Z}^+$ or, in the non-binary case, $P \to \{\mathbb{Z}^+, \mathbb{Z}^+ \times \mathbb{Z}^+ \times \mathbb{Z}^+\}$. How this works is explained later. Further if $A \subseteq X$ and for all sets $P_i \in P$ we have $A \cap P_i = \emptyset$ then A is said to *not cross* P.

When dealing with binary trees the following reductions were used:

1. The subtree reduction whose pseudocode is given in algorithm 3.2.1 and is illustrated in Figure 4.4. With the addition of w and P as detailed above one could throw away the leaves corresponding to elements that cross P and occur in the subtree being reduced, but it does not hurt anything other than a little space to leave them be.

2. A somewhat more complex chain reduction. For any chain $A = (a_0, a_1, \ldots, a_n)$ of length greater than or equal to three that occurs in $\mathcal{T}$ and $\mathcal{T}'$. Set $\mathcal{T}$ to be $\mathcal{T} \mid \overline{A - \{a_0, a_n\}}$ and likewise with $\mathcal{T}'$. Add $\{a_0, a_n\}$ to P with weight

$$w(a_0, a_n) = n - 2 + \sum_{\substack{\{a_i, a_j\} \in P \\ a_i, a_j \in \{a_1, \ldots, a_n\}}} w(a_i, a_j)$$

   and delete all pairs in P of the form $\{a_i, a_{i+1}\}$ for $i = 0, \ldots, n-1$ as illustrated in Figure 4.5 and formulated in Algorithm 4.1.1.

3. If A is a common pendant cluster of $\mathcal{T}$ and $\mathcal{T}'$ then

$$h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T} \mid A \cup \{\rho\}, \mathcal{T}' \mid A \cup \{\rho\}) + h(\mathcal{T} \mid (\overline{A} \cup \{\min A\}), \mathcal{T}' \mid (\overline{A} \cup \{\min A\}))$$

   as in Algorithm 4.1.2 and Figure 4.6.

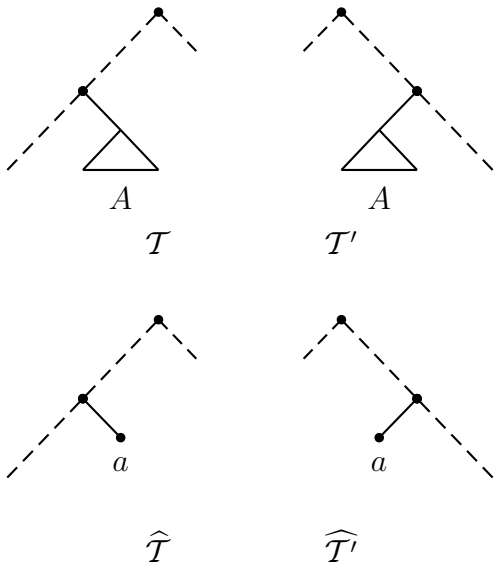


Figure 4.4: Two binary rooted phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with a common pendant subtree A and after the replacement with a gives $\mathcal{T}$ and $\mathcal{T}'$

Algorithm 4.1.1: HYBRID-CHAINREDUCTION$(\mathcal{T}, \mathcal{T}', w)$

procedure HYBRID-CHAINREDUCTION$(\mathcal{T}, \mathcal{T}', w)$
$(a_1, \ldots, a_n) \leftarrow$ maximal common chain of $\mathcal{T}$ and $\mathcal{T}'$
if $n \geqslant 3$
then
- $A \leftarrow \{a_1, \ldots, a_n\}$
- $w(a_1, a_n) \leftarrow n - 2 + \sum_{i=1}^{n-1} w(a_i, a_{i+1})$
- $w' \leftarrow \{w(a_1, a_n)\} \cup w$ restricted to pairs not in $\{a_1, \ldots, a_n\}$
- $\mathcal{T} \leftarrow \mathcal{T} \mid \overline{A - \{a_1, a_n\}}$
- $\mathcal{T}' \leftarrow \mathcal{T}' \mid \overline{A - \{a_1, a_n\}}$
- $(\mathcal{T}, \mathcal{T}', w) \leftarrow$ HYBRID-CHAINREDUCTION$(\mathcal{T}, \mathcal{T}', w')$

return $(\mathcal{T}, \mathcal{T}', w)$

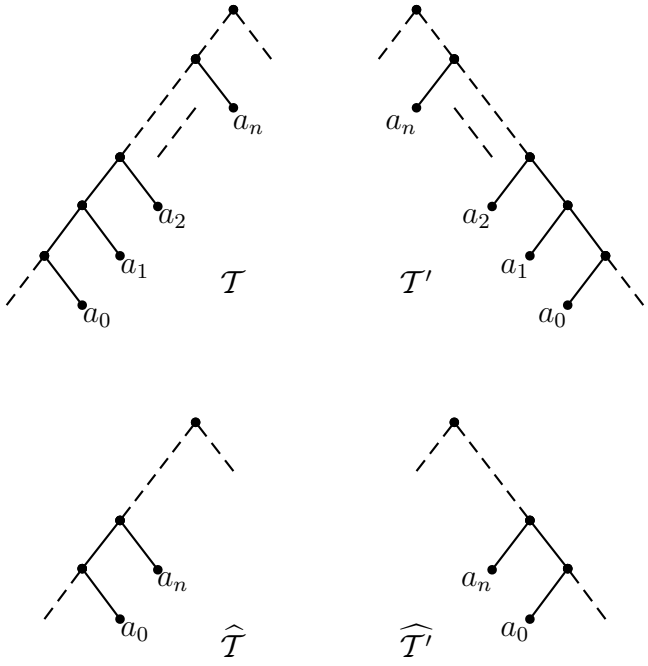


Figure 4.5: Chain Reduction for a pair of trees $\mathcal{T}$ and $\mathcal{T}'$ which gives $\widehat{\mathcal{T}}$ and $\widehat{\mathcal{T}'}$.

A further few aspects of agreement forests are now added in order to deal with the additional weighting.

Definition 4.4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be a pair of weighted binary phylogenetic trees with a common label set. An agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ is* legitimate *if it is acyclic and if $\{a, b\} \in P$ then either $\{\{a\}, \{b\}\} \subseteq \mathcal{F}$ or $\{a, b\}$ is a subset of a labelset of a tree in $\mathcal{F}$.*

Definition 4.5. *Let $\mathcal{F}$ be an agreement forest for weighted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with associated set P. The* weight *of $\mathcal{F}$ is given by*

$$w(\mathcal{F}) = (|\mathcal{F}| - 1) + \sum_{\substack{\{a,b\} \in P \\ a \text{ and } b \text{ isolated in } \mathcal{F}}} w(a, b)$$

Lastly we define $f(\mathcal{T}, \mathcal{T}')$ to be the minimum weight of a legitimate agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ and obtain the following result.

Lemma 4.6 (Theorem 1.1, Bordewich and Semple (2007b))**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be a pair of weighted binary phylogenetic trees with a common label set X. Let $\widehat{\mathcal{T}}$ and $\widehat{\mathcal{T}'}$ be the pair of weighted phylogenetic trees with label set $\widehat{X}$ obtained from $\mathcal{T}$ and $\mathcal{T}'$ by applying the subtree and chain*

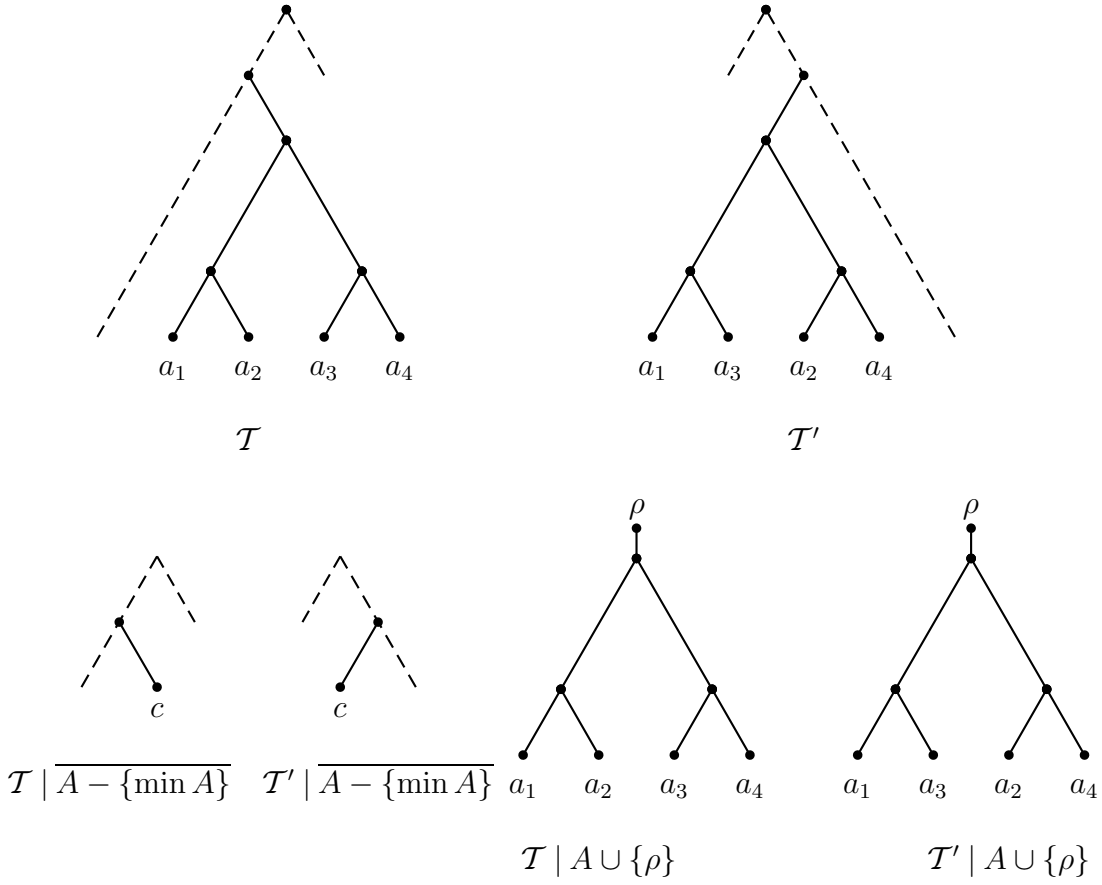


Figure 4.6: Cluster Reduction for a pair of phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$.

Algorithm 4.1.2: CLUSTERREDUCTION($\mathcal{T}, \mathcal{T}', w$)

procedure CLUSTERREDUCTION($\mathcal{T}, \mathcal{T}', w$)

$C \leftarrow$ minimal common cluster of $\mathcal{T}$ and $\mathcal{T}'$

$\mathcal{T}_1 \leftarrow \mathcal{T} \mid C \cup \{\rho\}$

$\mathcal{T}_2 \leftarrow \mathcal{T} \mid \overline{C - \{\min C\}}$

$\mathcal{T}_1' \leftarrow \mathcal{T}' \mid C \cup \{\rho\}$

$\mathcal{T}_2' \leftarrow \mathcal{T}' \mid \overline{C - \{\min C\}}$

$w_1 \leftarrow w$ restricted to pairs of taxa in C

$w_2 \leftarrow w$ restricted to pairs of taxa not in C

return $(\mathcal{T}_1, \mathcal{T}_1', w_1, \mathcal{T}_2, \mathcal{T}_2', w_2)$

reductions until no longer possible. Then

$$h(\mathcal{T}, \mathcal{T}') = f(\widehat{\mathcal{T}}, \widehat{\mathcal{T}'})$$

Further, after the reductions $|\widehat{X}| < 14\, h(\mathcal{T}, \mathcal{T}')$ *and the decision problem is solvable in* $\mathcal{O}((28k)^k + |\widehat{X}|^3)$ *where* $k = h(\mathcal{T}, \mathcal{T}')$.

An algorithm for this problem was formulated in Bordewich et al. (2007a) and is implemented in software available at `http://www.bi.uni-duesseldorf.de/~linz` and `http://www.math.canterbury.ac.nz/~cas83`.

As noted in section in the previous chapter phylogenetic trees are not always binary. Thus, in Linz and Semple (2008) the hybridisation number is extended to non binary trees.

Definition 4.7. *Let* $\mathcal{T}$ *and* $\mathcal{T}'$ *be two rooted phylogenetic trees with a common label set.*

$$h(\mathcal{T}, \mathcal{T}') = \min_{\substack{\mathcal{B} \text{ is a binary refinement of } \mathcal{T} \\ \mathcal{B}' \text{ is a binary refinement of } \mathcal{T}'}} h(\mathcal{B}, \mathcal{B}')$$

Considering the following decision problem

PROBLEM: Hybridisation

INSTANCE: Two rooted phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labelled by X and an integer k.

QUESTION: Is $h(\mathcal{T}, \mathcal{T}') \leqslant k$?

and the aims and results in the thesis thus far, it should be no surprise that the decision problem is NP-hard and fixed parameter tractable.

Theorem 4.8 (Linz and Semple (2008))**.** *Let* $\mathcal{T}$ *and* $\mathcal{T}'$ *be two rooted phylogenetic trees. then calculating* $h(\mathcal{T}, \mathcal{T}')$ *is NP-hard.*

Theorem 4.9 (Theorem 1.1, Linz and Semple (2008))**.** *The decision problem Hybridisation is fixed parameter tractable with* $h(\mathcal{T}, \mathcal{T}')$ *the parameter.*

In proving the above theorem the following reductions are used.

1. Subtree reduction as for the binary case.

2. Long Chain Reduction: Take a maximal chain $(a_1, a_2, \ldots, a_n)$ in $\mathcal{T}$ and $\mathcal{T}'$ with $n \geqslant 4$ with the properties

(a) The chain has at least three internal parents and at least three internal elements in $\mathcal{T}$ and $\mathcal{T}'$.

(b) If a_1 is external in one tree, then a_2 is internal in the same tree and a_1 is internal in the other tree.

(c) If a_n is external in one tree then a_{n-1} is in internal in the same tree, a_n is internal in the other tree and there are not exactly three internal parents of which one has a_n as its sole child in $\{a_1, a_2, \ldots, a_n\}$.

Then, depending on whether $\emptyset$, $\{a_1\}$, $\{a_n\}$ or $\{a_1, a_n\}$ are the elements external in $\mathcal{T}$ or $\mathcal{T}'$, replace the chain with (a, b, c), (e_1, a, b, c), (a, b, c, e_2) or (e_1, a, b, c, e_2) respectively such that $p_{\mathcal{T}}(e_1) \neq p_{\mathcal{T}}(a) = p_{\mathcal{T}}(b) \neq p_{\mathcal{T}}(c) \neq p_{\mathcal{T}}(e_2)$ and $p_{\mathcal{T}'}(e_1) \neq p_{\mathcal{T}'}(a) \neq p_{\mathcal{T}'}(b) = p_{\mathcal{T}'}(c) \neq p_{\mathcal{T}'}(e_2)$. Respectively the new set $\{a, b, c\}$, $\{e_1, a, b, c\}$, $\{a, b, c, e_2\}$ or $\{e_1, a, b, c, e_2\}$, call it S is added to P and to w the tuple of weights is added such that $w_1 = n - S$, w_2 is the number of internal parents in $\mathcal{T}$ minus the number of internal parents in the new chain, and w_3 is the number of internal parents in $\mathcal{T}'$ minus the number of internal parents in the new chain.

3. Short Chain Reduction: For $n \geqslant 3$ let $A = (a_1, a_2, \ldots, a_n)$ be a chain of both $\mathcal{T}$ and $\mathcal{T}'$ such that in one of the trees—say, $\mathcal{T}$—this chain has exactly one parent, while in the other tree—say, $\mathcal{T}'$—the chain has at least three internal parents and no external parents. Replace this chain in $\mathcal{T}$ and $\mathcal{T}'$ with the chain (a, b) such that $p_{\mathcal{T}}(a) = p_{\mathcal{T}}(b)$ and $p_{\mathcal{T}'}(a) \neq p_{\mathcal{T}'}(b)$. Add the new set $\{a, b\}$ to P and assign it weight $n - 2$.

4. Cluster Reduction is essentially as before except that vertex clusters are being considered.

How these reductions come in to play with the final agreement forest will not be discussed here and the interested reader is referred to Linz and Semple (2008). Note, however, that a legitimate agreement forest in the non-binary situation has more restrictions on how the sets in P are separated.

The rest of this chapter presents two algorithms whose results are in Table 4.1. The first WEED is somewhat faster than the former PERL implementation of the algorithm in Bordewich et al. (2007a). The last two, both based on HYBRID-REKERNELISE are based on the same rekernelisation technique however one restricts the options one may consider when removing

leaves and the other demonstrates the running time if this restriction is dropped. The rekernelising with the additional ordering on which subtrees one may remove gives truly stunning results; results that formerly took over two days take little over ten minutes!

4.2 Weeds

A natural place to start in the search of maximum acyclic agreement forests is to look at the complementary problem. Instead of asking *what is the maximum agreement forest?* consider *what is the minimum non-agreement forest?* It may come as a surprise that, for a forest $\mathcal{F}$ of $\mathcal{T}$ that is an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ obtained by deleting edges of $\mathcal{T}$, the minimum forbidden subforests—that is, forests that are forests of $\mathcal{T}$ but not $\mathcal{T}'$ and have the most elements—can be expressed as either pairs of trees each containing two labels from $\mathcal{L}(\mathcal{T})$; single trees containing three labels from X; or sets of trees each containing two labels in X. This concept immediately brings to mind the restricted minors that exist elsewhere in combinatorics, except that they depend on the trees in question. In light of the fact that they are so small, and prevent trees in agreement forests from growing large, the set of these sets of trees will be called the *weeds of $\mathcal{T}$ with respect to $\mathcal{T}'$* denoted $\mathcal{W}_{\mathcal{T}}(\mathcal{T}')$.

Lemma 4.10. *If v is the most common recent ancestor of some set of leaves $X' \subseteq X$ of a tree $\mathcal{T}$, then $v = \mathrm{mrca}_{\mathcal{T}}\{a, b\}$ for some $a, b \in X'$.*

Proof. Trivially, take a from the descendants of the left child of v and b from the descendants of the right child. Then $v = \mathrm{mrca}_{\mathcal{T}}\{a, b\}$. □

Lemma 4.11. *If $\mathcal{F} = \{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n\}$ is a forest of $\mathcal{T}$ and a cyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ then there is a forest $\mathcal{F}' = \{\mathcal{T}_0', \mathcal{T}_1', \ldots, \mathcal{T}_n'\}$ such that for each $i \in \{0, 1, \ldots, n\}$ we have $\mathcal{T}_i'$ is a subtree of $\mathcal{T}_i$, $|\mathcal{L}(\mathcal{T}_i')| = 2$ and $\mathcal{F}'$ is cyclic in $\mathcal{T}$ and $\mathcal{T}'$.*

Proof. Let $\mathcal{C}$ be a cycle in $G_{\mathcal{F}}$. For each tree $\mathcal{T}_i$ in $\mathcal{C}$ create $\mathcal{T}_i'$ by removing all except two leaves that are descendants of distinct children of $\mathcal{T} \mid \mathcal{L}(\mathcal{T}_i)$ and contract. By construction $\mathrm{mrca}_{\mathcal{T}}\, \mathcal{L}(\mathcal{T}_i) = \mathrm{mrca}_{\mathcal{T}}\, \mathcal{L}(\mathcal{T}_i')$ and similarly for $\mathcal{T}'$. Since this is the case the set consisting of $\{\mathcal{T}_0', \mathcal{T}_1', \ldots, \mathcal{T}_n'\}$ is also cyclic. □

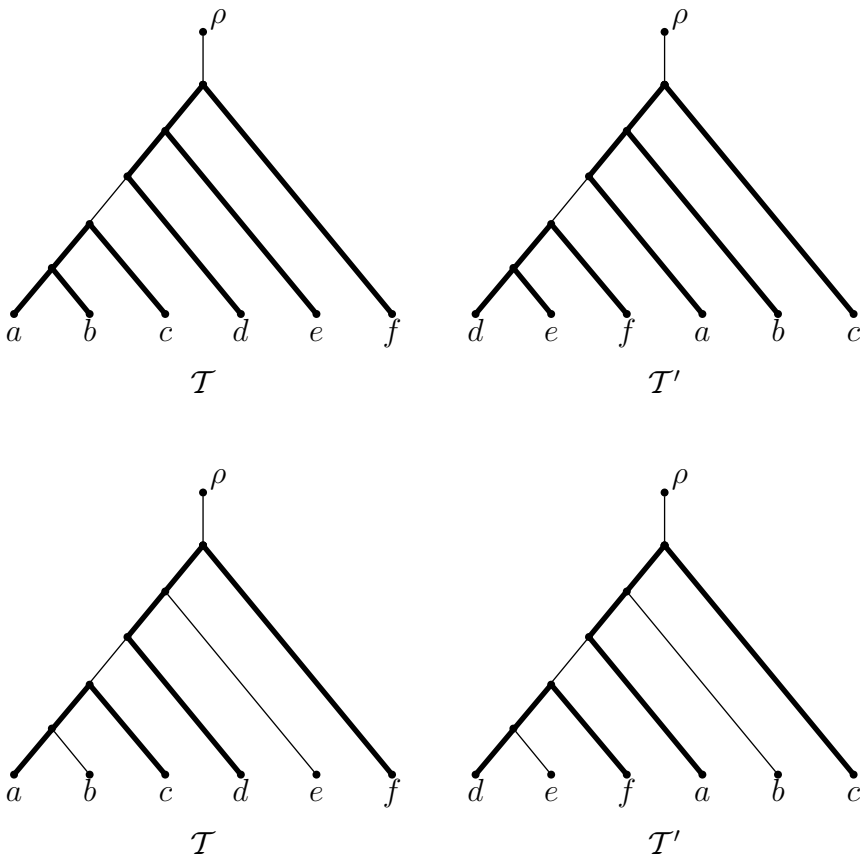


Figure 4.7: An illustration of lemma 4.11. where $\mathcal{F} = \{\{a, b, c\}, \{d, e, f\}, \{\rho\}\}$ is cyclic and contains cyclic trees $\mathcal{T} \mid \{a, c\}$ and $\mathcal{T} \mid \{d, f\}$.

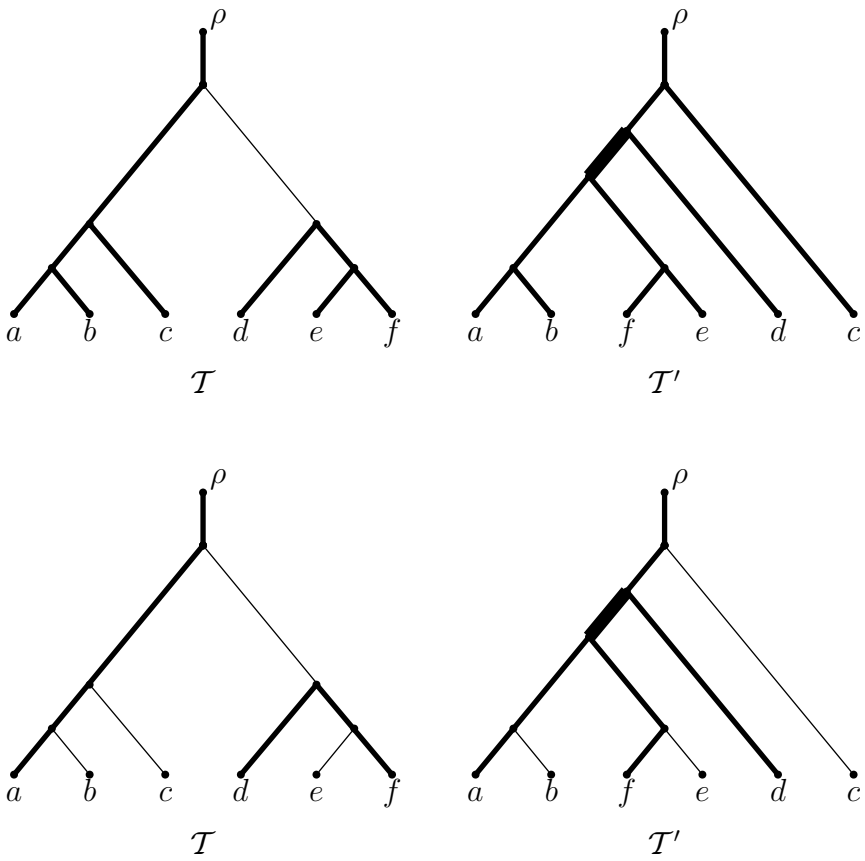
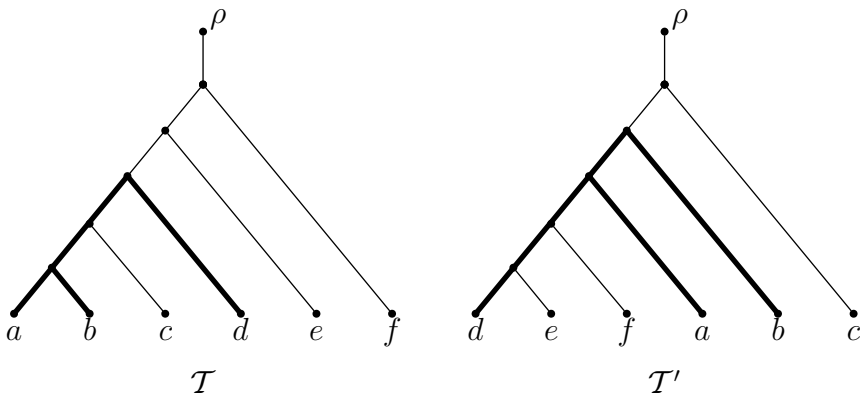


Figure 4.8: An illustration of lemma 4.12. where $\mathcal{F} = \{\{\rho, a, b, c\}, \{d, e, f\}, \{\rho\}\}$ is not vertex disjoint in $\mathcal{T}'$ and contains similarly non vertex disjoint trees $\mathcal{T} \mid \{\rho, a\}$ and $\mathcal{T} \mid \{d, f\}$.



$\mathcal{T}$ $\mathcal{T}'$

Figure 4.9: An illustration of lemma 4.13. where $\mathcal{F} = \{\{\rho, a, b, c, d, e, f\}\}$ is not in agreements and contains non agreement tree $\mathcal{T} \mid \{a, b, d\}$ (as well as many others).

Lemma 4.12. *If $\mathcal{F}$ is a forest of $\mathcal{T}$ and is not an edge disjoint forest of $\mathcal{T}'$ then it contains $\mathcal{T} \mid \{a, b\}$ and $\mathcal{T} \mid \{c, d\}$ for distinct $a, b, c, d \in \mathcal{L}(\mathcal{T})$ such that $\mathcal{T} \mid \{a, b\}$ and $\mathcal{T} \mid \{c, d\}$ are not vertex disjoint subtrees of $\mathcal{T}'$.*

Proof. In $\mathcal{T}'$ there must be at least two trees $\mathcal{T}_i$ and $\mathcal{T}_j$ that must share at least two vertices and an edge. Set $e = (u, v)$ to be the edge between the two vertices that is in conflict in $\mathcal{T}'$ due to $\mathcal{T}_i$ and $\mathcal{T}_j$ in the forest. Now take $a \in \mathcal{C}_{\mathcal{T}'|\mathcal{L}(\mathcal{T}_i)}(e) \cap \mathcal{L}(\mathcal{T}_i)$ and $b \in \mathcal{L}(\mathcal{T}_i) - \mathcal{C}_{\mathcal{T}'|\mathcal{L}(\mathcal{T}_i)}(e)$. In words we have a as a descendant of e and b as some vertex that will cause e to remain after we contract edges since $\mathrm{mrca}_{\mathcal{T}'}\{a, b\} \rightsquigarrow u \rightsquigarrow v \rightsquigarrow a$. So now $e \in E(\mathcal{T}'(\{a, b\}))$ and $\mathcal{T}' \mid \{a, b\} = \mathcal{T}_i \mid \{a, b\} \subset \mathcal{T}_i$.

Similarly we may find that $e \in E(\mathcal{T}'(\{c, d\}))$ via $c \in \mathcal{C}_{\mathcal{T}'|\mathcal{L}(\mathcal{T}_j)}(e) \cap \mathcal{L}(\mathcal{T}_j)$ and $b \in \mathcal{L}(\mathcal{T}_j) - \mathcal{C}_{\mathcal{T}'|\mathcal{L}(\mathcal{T}_j)}(e)$. Thus $\mathcal{T} \mid \{a, b\}$ and $\mathcal{T} \mid \{c, d\}$ are not vertex disjoint in $\mathcal{T}'$ and the proof is complete. □

Lemma 4.13. *If $\mathcal{F}$ is a forest of $\mathcal{T}$ but not a forest for $\mathcal{T}'$ then it contains a tree $\mathcal{T} \mid \{a, b, c\}$ for distinct $a, b, c \in X$ such that $\mathcal{T} \mid \{a, b, c\}$ is not a subtree of $\mathcal{T}'$.*

Proof. Take a tree $\mathcal{T}_i \in \mathcal{F}$ such that $\mathcal{T}_i$ is not a subtree of $\mathcal{T}'$. Find the minimal cluster in $\mathcal{T}_i$ that is also a cluster of $\mathcal{T}' \mid \mathcal{L}(\mathcal{T}_i)$ but not a subtree. Take a in the cluster of one of the children of the root of $\mathcal{T}_i$ and b in the cluster of the other child. All that remains now is to show the existence of a leaf c that is in the cluster containing a in $\mathcal{T}$ and in the cluster containing b in $\mathcal{T}'$. Assume that

such a leaf does not exist, this means that $\mathcal{C}_{\mathcal{T}|\mathcal{L}(\mathcal{T}_i)}(a) = \mathcal{C}_{\mathcal{T}'|\mathcal{L}(\mathcal{T}_j)}(a)$ and similarly for b. This contradicts $\mathcal{T}_i$ being a minimal cluster however, thus c exists. □

Theorem 4.14. *A forest $\mathcal{F}$ that is a forest of $\mathcal{T}$ is an agreement forest for trees $\mathcal{T}$ and $\mathcal{T}'$ if and only if*

- *for every $\mathcal{T}_i \mid \{a, b\}$ and $\mathcal{T}_j \mid \{c, d\}$, with distinct $a, b, c, d \in \mathcal{L}(\mathcal{T})$ we have $\mathcal{T}_i \mid \{a, b\}$ and $\mathcal{T} \mid \{c, d\}$ edge disjoint in $\mathcal{T}'$*
- *every $\mathcal{T}_i \mid \{a, b, c\}$ for distinct $a, b, c \in X$ and every $\mathcal{T}_i \in \mathcal{F}$ we have $\mathcal{T} \mid \{a, b, c\}$ is a subtree of $\mathcal{T}'$,*
- *and for every set of trees $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n\} \in \mathcal{F}$ we have that the set consisting of every subtree $\mathcal{T}_i'$ of $\mathcal{T}_i$ for $Ii \in \{0, 1, \ldots, n\}$ with only two leaves $\{\mathcal{T}_0', \mathcal{T}_1', \ldots, \mathcal{T}_n'\}$ is acyclic.*

Proof. Follows from the above lemmas. □

With the weeding results under our belts there is a further comment to make—namely, we not only wish that the lemmas devised hold, but that the set we arrive at are subtrees of $\mathcal{T}$. The method used to accomplish this was to devise a mapping that takes as input a rooted binary tree $\mathcal{T}$ and string of zeros and ones, or bit string, and produces a forest of $\mathcal{T}$. The way this works is that each edge of $E(\mathcal{T})$ is identified with a place in the bit string. As such the bit string is in $\mathbb{Z}_2^{|E(\mathcal{T})|}$ and additionally every element in $\mathbb{Z}_2^{|E(\mathcal{T})|}$ that is equal to 0 corresponds to a deletion of an edge in $\mathcal{T}$, although after contraction two different bit strings may generate isomorphic forests, see lemma 4.18 for more about how to avoid this.

Definition 4.15. *Let $E = (e_1, e_2, \ldots, e_n)$ be an ordered set of edges of a tree $\mathcal{T}$. Let the mask $\mathfrak{m}$ be a bit string of length $|E|$. Then $\mathfrak{m}(\mathcal{T})$ is the forest of $\mathcal{T}$ with the edge set $\{e_i : e_i \in E$ and the ith entry of $\mathfrak{m}$ is $1\}$. Let $\mathfrak{m}[i]$ denote the ith entry of $\mathfrak{m}$ when considered as a bit string.*

Other nice aspects of this method is that some deletion of edges will only appear once and that if some element $b \in \mathbb{Z}_2^{|E(\mathcal{T})|}$ is an agreement forest then any other bit string with zeroes where b has zeros will also yield an agreement forest and that if b is a string that is not an agreement forest then any bit string with ones in the same place is also not an agreement forest. Finally this enables us to search for an agreement forest using two trees and a bit string, removing some of the stress other algorithms seeming to place on computer memory.

Definition 4.16. *Let $\mathfrak{m}$ and $\mathfrak{m}'$ be two masks. If the ith entry of $\mathfrak{m}'$ is 0 whenever the ith entry of $\mathfrak{m}$ is 0 then define $\mathfrak{m}' \subseteq \mathfrak{m}$.*

Corollary 4.17. *$\mathfrak{m}' \subseteq \mathfrak{m}$ if and only if $\mathfrak{m}'(\mathcal{T})$ is a subforest of $\mathfrak{m}(\mathcal{T})$.*

Proof. By definition $E(\mathfrak{m}(\mathcal{T})) \subseteq E(\mathfrak{m}(\mathcal{T}'))$. The result trivially follows. □

For each of the weeds of $\mathcal{T}$ with respect to $\mathcal{T}'$, the bit string whose output corresponds to the weed is formed and then an algorithm is used that removes 1s from a bit string. When a bit string that contains none of the weeds is found an agreement forest has also been found. If the number of zeros in the bit string is counted, this corresponds to the number of edges deleted and thus the hybridisation number to obtain the acyclic agreement forest. Thus the bit strings with the smallest number of zeroes that contain none of the bit strings that correspond to weeds are the bit strings that yield the maximum agreement forests. Naturally one has to take a little more care with weighted forests.

In the context of the weeding algorithm, the mapping between elements of the bit string and the edges of $\mathcal{T}$ are not so important—although there may well be an optimal way of picking them.

Lemma 4.18. I*f $\mathcal{F}$ is a forest obtained from $\mathcal{T}$ by deleting edges, but neither suppressed nor contracted, that contains a vertex v such that $d^+(v) = 1$ but $d^-(v) = 0$ then there is another forest $\mathcal{F}'$ such that $\mathcal{F} \cong \mathcal{F}'$ after suppressing edges and vertices that results from the same number of edge deletions but without any disconnected vertices of out-degree one prior to suppression and contraction.*

Proof. Let v be such a vertex. If we take the arcs $(u, v), (v, w) \in A(\mathcal{T})$ such that $(v, w) \in A(\mathcal{F})$. If we remove (v, w) and add (u, v) we clearly get a forest that is isomorphic to $\mathcal{F}$ after forced contractions since in the former case (v, w) will be removed, not having being attached to anything with labels on its head, and in the latter case (u, v) will be removed, not having any labels in the cluster of its tail.

The next step is to show that the above operation causes the number of such vertices to reduce by at least one. So consider u, there are three options for the edges u has previously: it has one in edge, it has one out edge or it has both. We shall discount the possibility that it is completely disconnected because if such a vertex exists then we already know that it is not a maximum agreement forest, nor contains one, due to the result in Linz (2008). If u has an

in-edge then clearly adding (u, v) will not add an additional vertex with only an out-edge. If u has a single out-edge then adding (u, v) will in fact cause u to fail to be a vertex with a single out-edge so the total number of such vertices will drop by two □

The motivation for the above proof is that when going through the bit string possible search paths should be pruned from the search tree as soon as possible. Our current search path pruning options at this point are occur if the forest has an isolated internal vertex (or root) and if the forest has a vertex with total and out-degree one.

A few additional notes serve to be made about the algorithm WEED detailed in Algorithm 4.2.1. For good performance it seems advantageous to calculate the set of weeds before running WEED, however it is likely that the set of weeds takes exponential space in terms of the hybridisation number. Thus a certain structure was used to not only minimise this, but also enable us to check the set faster than simply scanning down a list of items.

First the minimal set of edges required for a non-acyclic maximum agreement forest was determined and the corresponding mask $\mathfrak{m}$ found. A initially empty binary search tree was then augmented by traversing $\mathfrak{m}$ entry by entry until the search tree had no such path, then inserting the remaining values of $\mathfrak{m}$ as a linked list of values. The original idea was to construct a DFA of the weeds so that it could be checked in linear time. A DFA is a type of automaton that is capable of recognising strings that match a certain pattern, for a more thorough description see Hopcroft, Motwani, and Ullman (2001). However, other than there seemingly being no present results that would easily allow me to build the required structures, each mask corresponds to a number of additions, since when comparing we not only want ones where ones are but *either* ones or zeroes at every other entry, this could potentially result in a large DFA for few weeds.

There is an extension of this method to the non-binary problem. Clearly the formulated method cannot be applied as is, since the problem does not come down to removing edges. However if in place of a mask on the edges, an array of the same length of the label set consisting of numbers ranging between 1 and the size of the label set could be used. In this case two labels would be considered to be in the same label set if they share the same number in the array.

4.3 Rekernelisation

The hybridisation number rekernelisation technique is a little different to the previously discussed rooted subtree prune and regraft rekernelisation technique and many of the following

Algorithm 4.2.1: WEED($\mathcal{T}, \mathcal{T}'$)

procedure WEEDING($\mathcal{T}, \mathcal{T}', w, k, \mathfrak{m}$)
if $k \leqslant 0$
  then return (k)
if any vertex is isolated in $\mathfrak{m}(\mathcal{T})$ **or** $\exists v \in V(\mathfrak{m}(\mathcal{T})) : d^-(v) = 0, d^+(v) = 1$
  then return (k)
if there exists a $\mathfrak{m}' \in \mathcal{W}_{\mathcal{T}}(\mathcal{T}')$ **such that** $\mathfrak{m}' \subseteq \mathfrak{m}$
  then
    for $i =$ **position of rightmost zero in** $\mathfrak{m}$ **to length of** $\mathfrak{m}$
    do
      $e \leftarrow$ The edge corresponding to the ith entry in $\mathfrak{m}$
      if e **is adjacent to a label** a **such that** $\{a, b\} \in P$
      then
        $j \leftarrow$ the index of $\mathfrak{m}$ corresponding to edge adjacent to b
        if $a < b$
        then
          $\mathfrak{m}[i] \leftarrow 0$
          $\mathfrak{m}[j] \leftarrow 0$
          $k \leftarrow \min\{k, \text{WEEDING}(\mathcal{T}, \mathcal{T}', w, k - (w\{a, b\} + 2), \mathfrak{m}) + w\{a, b\} + 2\}$
          $\mathfrak{m}[i] \leftarrow 1$
          $\mathfrak{m}[j] \leftarrow 1$
      else
        $\mathfrak{m}[i] \leftarrow 0$
        $k \leftarrow \min\{k, \text{WEEDING}(\mathcal{T}, \mathcal{T}', w, k - 1, \mathfrak{m}) + 1\}$
        $\mathfrak{m}[i] \leftarrow 1$
  else $k \leftarrow 0$
return (k)

main
$k \leftarrow |\mathcal{L}(\mathcal{T})|$
$(\mathcal{T}, \mathcal{T}') \leftarrow$ SUBTREEEREDUCTION$(\mathcal{T}, \mathcal{T}')$
$(\mathcal{T}, \mathcal{T}', w) \leftarrow$ HYBRID-CHAINREDUCTION$(\mathcal{T}, \mathcal{T}', w)$
$C \leftarrow$ Labels of a minimal common cluster of $\mathcal{T}$ and $\mathcal{T}'$
if $1 < |C| < |\mathcal{L}(\mathcal{T})|$
  then
    $(\mathcal{T}_1, \mathcal{T}_1', w_1, \mathcal{T}_2, \mathcal{T}_2', w_2) \leftarrow$ CLUSTERREDUCTION$(\mathcal{T}, \mathcal{T}', w)$
    $k' \leftarrow$ WEEDING$(\mathcal{T}_1, \mathcal{T}_1', w_1, k,$ bit string of 1s of length $|\mathcal{L}(\mathcal{T}_1)|)$
    return $(k' +$ WEED$(\mathcal{T}_2, \mathcal{T}_2', w_2, k - k',$ bit string of 1s of length $|\mathcal{L}(\mathcal{T}_2)|))$
return (WEED$(\mathcal{T}, \mathcal{T}', w, k,$ bit string of 1s of length $|\mathcal{L}(\mathcal{T})|)$)

results have been submitted as Collins et al. (submitted). It was first pointed out by Simone Linz who observed that $G_{\mathcal{F}}$ had vertices of out-degree zero that corresponded to pendant subtrees, thus how the trees were broken up had a restriction added. Further, that if these were removed then it may be possible to rekernelise the problem resulting in a faster algorithm.

Lemma 4.19. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unweighted phylogenetic trees with leaves labelled by X. Then for some pendant subtree with leaves labelled by A*

$$1 + m_a(\mathcal{T} \mid \overline{A}, \mathcal{T}' \mid \overline{A}) \geqslant m_a(\mathcal{T}, \mathcal{T}')$$

Proof. Let $\mathcal{F}_A$ be a maximum acyclic agreement forest for $\mathcal{T} \mid \overline{A}$ and $\mathcal{T}' \mid \overline{A}$ of minimum weight. It then follows that

$$\mathcal{F} = \mathcal{F}_A \cup \{\mathcal{T} \mid A\}$$

is an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Additionally $|\mathcal{F}| = |\mathcal{F}_A| + 1$ and so

$$\begin{aligned} m_a(\mathcal{T} \mid \overline{A}, \mathcal{T}' \mid \overline{A}) + 1 &= |\mathcal{F}_A| - 1 + 1 \\ &= |\mathcal{F}| \\ &\geqslant m_a(\mathcal{T}, \mathcal{T}') \end{aligned}$$

□

Corollary 4.20. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unweighted phylogenetic trees with leaves labelled by X with no common pendant subtree whose leaf set size is at least 2. Then for each leaf $\ell \in X$*

$$1 + m_a(\mathcal{T} \mid \overline{\{\ell\}}, \mathcal{T}' \mid \overline{\{\ell\}}) \geqslant m_a(\mathcal{T}, \mathcal{T}')$$

Lemma 4.21. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unweighted non-isomorphic phylogenetic trees with leaves labelled by X. Then there exists a common pendant subtree with leaves labelled by A such that*

$$m_a(\mathcal{T}, \mathcal{T}') = m_a(\mathcal{T} \mid \overline{A}, \mathcal{T}' \mid \overline{A}) + 1$$

Proof. Let $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \ldots, \mathcal{T}_k\}$ be a maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Since $G_{\mathcal{F}}$ is acyclic there must be a vertex $\mathcal{T}_i$ in $G_{\mathcal{F}}$ such that the out-degree of $\mathcal{T}_i$ in $G_{\mathcal{F}}$ is zero. Since for all $j \in \{1, \ldots, k\}$ we know that $\operatorname{mrca} \mathcal{L}(\mathcal{T}_\rho) \rightsquigarrow \operatorname{mrca} \mathcal{L}(\mathcal{T}_j)$ in both trees it follows that $i \neq \rho$ and thus set $A = \mathcal{L}(\mathcal{T}_i) \subseteq X$.

Since $\mathcal{F}$ is a maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ we have $\mathcal{F}_A = \mathcal{F} - \{\mathcal{T}_i\}$ is an acyclic agreement forest for $\mathcal{T} \mid \overline{A}$ and $\mathcal{T}' \mid \overline{A}$. Further $|\mathcal{F}_A| = |\mathcal{F}| - 1$ and so

$$\begin{aligned} m_a(\mathcal{T}, \mathcal{T}') &= |\mathcal{F}| - 1 \\ &= |\mathcal{F}_A| \\ &\geqslant m_a(\mathcal{T} \mid \overline{A}, \mathcal{T}' \mid \overline{A}) + 1 \end{aligned}$$

This combined with the inequality in Lemma 4.19 gives the required result. □

Corollary 4.22. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unweighted phylogenetic trees with leaves labelled by X with no common pendant subtree whose leaf set size is at least 2. Then there exists some leaf $\ell \in X$ such that*

$$1 + m_a(\mathcal{T} \mid \overline{\{\ell\}}, \mathcal{T}' \mid \overline{\{\ell\}}) = m_a(\mathcal{T}, \mathcal{T}')$$

Those who now read Collins et al. (submitted) will doubtless be wondering why these pieces of the lemmas have been separated out. There is more than one reason. The first is that the above results place no restrictions as to whether or not the phylogenetic trees are binary which allows the corresponding results to chain reductions to be done separately. The other is an upcoming result where we apply an ordering to an acyclic agreement forest for two trees which can be viewed as an order based on the forests created as the rekernelising technique is used.

Lemma 4.23. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two weighted rooted binary phylogenetic trees with leaves labelled by X with no common pendant subtree whose leaf set size is at least 2, and let P be the disjoint collection of 2-element subsets of X associated with $\mathcal{T}$ and $\mathcal{T}'$. Then, for each $\ell \in X$, either*

$$2 + w(\ell, \ell') + f(\mathcal{T} \mid \overline{\{\ell, \ell'\}}, \mathcal{T}' \mid \overline{\{\ell, \ell'\}}) \geqslant f(\mathcal{T}, \mathcal{T}')$$

if ℓ crosses P for some other $\ell' \in X$, otherwise

$$1 + f(\mathcal{T} \mid \overline{\{\ell\}}, \mathcal{T}' \mid \overline{\{\ell\}}) \geqslant f(\mathcal{T}, \mathcal{T}')$$

Proof. Assume ℓ crosses P such that $\{\ell, \ell'\} \in P$. Let $\mathcal{F}_\ell$ be a legitimate agreement forest for $\mathcal{T} \mid \{\ell, \ell'\}$ and $\mathcal{T}' \mid \{\ell, \ell'\}$ of minimum weight. It then follows

$$\mathcal{F} = \mathcal{F}_\ell \cup \{\{\ell\}, \{\ell'\}\}$$

is such a forest for $\mathcal{T}$ and $\mathcal{T}'$. Additionally $|\mathcal{F}| = |\mathcal{F}_\ell| + 2$ and

$$\sum_{\substack{\{a,b\} \in P - \{\ell, \ell'\} \\ \{\{a\},\{b\}\} \subseteq \mathcal{F}_\ell}} w(a, b) = \sum_{\substack{\{a,b\} \in P \\ \{\{a\},\{b\}\} \subseteq \mathcal{F}}} w(a, b) - w(\ell, \ell')$$

Thus

$$\begin{aligned} 2 + w(\ell, \ell') + f(\mathcal{T} \mid \overline{\{\ell, \ell'\}}, \mathcal{T}' \mid \overline{\{\ell, \ell'\}}) &= |\mathcal{F}_\ell| + 1 + \sum_{\substack{\{a,b\} \in P - \{\ell, \ell'\} \\ \{\{a\},\{b\}\} \subseteq \mathcal{F}_\ell}} w(a,b) + w(\ell, \ell') \\ &= |\mathcal{F}| - 1 + \sum_{\substack{\{a,b\} \in P \\ \{\{a\},\{b\}\} \in \mathcal{F}}} w(a,b) \\ &\geqslant f(\mathcal{T}, \mathcal{T}') \end{aligned}$$

Lemma 4.19 covers the case in which ℓ does not cross P and both inequalities establish the result. □

Lemma 4.24. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two weighted rooted binary phylogenetic trees with leaves labelled by X with no common pendant subtree whose leaf set size is at least 2, and let P be the disjoint collection of 2-element subsets of X associated with $\mathcal{T}$ and $\mathcal{T}'$. Then there exists an element $\ell \in X$ such that*

$$f(\mathcal{T}, \mathcal{T}') = 2 + w(\ell, \ell') + f(\mathcal{T} \mid \overline{\{\ell, \ell'\}}, \mathcal{T}' \mid \overline{\{\ell, \ell'\}})$$

if ℓ crosses P, otherwise

$$f(\mathcal{T}, \mathcal{T}') = 1 + f(\mathcal{T} \mid \overline{\{\ell\}}, \mathcal{T}' \mid \overline{\{\ell\}})$$

Proof. Let $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ be a legitimate agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ of minimum weight. First observe that, as $G_\mathcal{F}$ is acyclic, it has a vertex $\mathcal{T}_i$ with $i \in \{\rho, 1, 2, \ldots, k\}$ whose out-degree is zero. Furthermore, since $\mathcal{T}$ and $\mathcal{T}'$ do not have any pendant subtree in common whose leaf set size is at least 2, $\mathcal{T}_i$ is a singleton in $\mathcal{F}$. Due to Lemma 1 of Baroni et al. (2006) ρ is never a singleton in $\mathcal{F}$. Therefore, we may assume that $\mathcal{L}(\mathcal{T})_i = \{\ell\}$ where $\ell \in X$.

First assume that ℓ crosses P in an element $\{\ell, \ell'\}$. Since $\mathcal{F}$ is a legitimate agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, we have $\{\ell'\} \in \mathcal{F}$ and, hence,

$$\mathcal{F}' = \mathcal{F} - \{\{\ell\}, \{\ell'\}\}$$

is a legitimate agreement forest for $\mathcal{T} \mid \overline{\{\ell, \ell'\}}$ and $\mathcal{T}' \mid \overline{\{\ell, \ell'\}}$. Further, we have $|\mathcal{F}| = 2 + |\mathcal{F}'|$ and

$$\sum_{\substack{\{a,b\} \in P \\ \{\{a\},\{b\}\} \in \mathcal{F}}} w(a,b) = \sum_{\substack{\{a,b\} \in P - \{\ell, \ell'\} \\ \{\{a\},\{b\}\} \subseteq \mathcal{F}'}} w(a,b) + w(\ell, \ell')$$

It then follows that

$$\begin{aligned} f(\mathcal{T},\mathcal{T}') &= |\mathcal{F}| - 1 + \sum_{\substack{\{a,b\}\in P \\ \{\{a\},\{b\}\}\in\mathcal{F}}} w(a,b) \\ &= 2 + |\mathcal{F}'| - 1 + \sum_{\substack{\{a,b\}\in P-\{\ell,\ell'\} \\ \{\{a\},\{b\}\}\subseteq\mathcal{F}'}} w(a,b) + w(\ell,\ell')) \\ &\geqslant 2 + w(\ell,\ell') + f(\mathcal{T} \mid \overline{\{\ell,\ell'\}}, \mathcal{T}' \mid \overline{\{\ell,\ell'\}}) \end{aligned}$$

The option where ℓ does not cross P is covered by Lemma 4.21. Lastly, note that for $\ell \in X$ the inequalities from Lemma 4.24 complete the inequality. □

Returning to non-binary chain reduction it is worth pointing out there is a small problem with scaling to multiple trees using this method. Concentrating on the a, b and c elements consider that the replacement is in order to tell when the elements of one tree need not be disconnected. This would mean for, say three trees $\mathcal{T}$, $\mathcal{T}'$ and $\mathcal{T}''$ a chain (a,b,c,d) would need a replacement such that $p_{\mathcal{T}}(a) = p_{\mathcal{T}}(b) \neq p_{\mathcal{T}}(c) \neq p_{\mathcal{T}}(d)$, $p_{\mathcal{T}'}(a) \neq p_{\mathcal{T}'}(b) = p_{\mathcal{T}'}(c) \neq p_{\mathcal{T}'}(d)$ and $p_{\mathcal{T}''}(a) \neq p_{\mathcal{T}''}(b) \neq p_{\mathcal{T}''}(c) = p_{\mathcal{T}''}(d)$. Further, in the case of $\mathcal{P} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ a chain of length $k+1$ would be required.

Definition 4.25. *Let $A = \{a_1, a_2, \ldots, a_n\}$ be a chain in a phylogenetic tree $\mathcal{T}$. The* parent partition $p_{\mathcal{T}}(A)$ of A in $\mathcal{T}$ *is the partition set A such that for any $A_i \in p_{\mathcal{T}}(A)$ and any $a \in A_i$ then $p_{\mathcal{T}}(a) = p_{\mathcal{T}}(b)$ for $b \in A$ if and only if $b \in A_i$.*

Lemma 4.26. *Let $\mathcal{T}$ and $\mathcal{T}'$ be phylogenetic trees with a common label set with no common subtrees and a maximal common chain $A = (a_1, a_2, \ldots, a_n)$ such that a_1 and a_n are external in at least one, but not both trees. If the parent partition of A in one of the trees is a subforest of a maximum agreement forest $\mathcal{F}$ then at some stage in the rekernelising algorithm the search path that corresponds to $\mathcal{F}$ we have A as a pendant subtree in the other tree.*

Proof. Assume that the parent partition of A with regards to $\mathcal{T}$ is a subforest of $\mathcal{F}$. We know that when rekernelising, items are added to the agreement forest if and only if they are pendant at some point in the process. Since we have assumed that there are no pendant subtrees, we know that any set of labels within the parent partition of A form a binary chain in $\mathcal{T}'$ where each end is possibly external. This in turn implies that each member of the parent partition of A with respect to $\mathcal{T}$ is pendant at some point in the rekernelising algorithm in both $\mathcal{T}$, which it is already, and $\mathcal{T}'$. The result then follows. □

The new chain reduction from the above result will still be applied in the same circumstances as the old long and short chain reductions. The intention is that this reduction scales better to multiple trees rather than being particularly more applicable. It shall be stated for the case of two trees and comments about how one might extend it to a more general setting will be discussed.

Let $A = (a_1, a_2, \ldots, a_n)$ be a chain with $n \geqslant 3$ such that

- The chain has either at least three internal parents in $\mathcal{T}$ and $\mathcal{T}'$ and at least three internal elements internal in $\mathcal{T}$ and $\mathcal{T}'$ or has exactly one parent.
- If a_1 is external in one tree then a_2 is internal in the same tree and a_1 is internal in the other tree if the chain in the other tree has more than one parent.
- If a_n is external in one of the trees, then a_{n-1} is internal in the same tree and a_n is internal in the other tree if the chain has more than one parent.

Depending on whether $\emptyset$, $\{a_1\}$, $\{a_n\}$ or $\{a_1, a_n\}$ is external in a chain with more than one parent in $\mathcal{T}$ or $\mathcal{T}'$ then respectively delete $A - \{a_1, a_n\}$, $A - \{a_1, a_2, a_n\}$, $A - \{a_1, a_{n-1}, a_n\}$, $A - \{a_1, a_2, a_{n-1}, a_n\}$. As before w now needs three values associated. Respectively add $\{a_1, a_n\}$,$\{a_2, a_n\}$,$\{a_1, a_{n-1}\}$ or $\{a_2, a_{n-1}\}$ to P and associate to w_1 the value n minus 2, 3, 3 or 4 respectively. To w_2 the number of internal parents in the original chain in $\mathcal{T}$ minus the number of internal elements in the new chain in $\mathcal{T}$. Lastly to w_3 the same as the second but with respect to $\mathcal{T}'$.

Now when rekernelising upon detaching an element that crosses P whether the pair $\{a, b\}$ is pendant in neither tree, just $\mathcal{T}$ or just $\mathcal{T}'$ (the other possibility is moot since it will be a common pendant subtree and removable via subtree reduction) is checked. Either way the other member of the chain is also detached and w_1, w_2 or w_3 is added respectively.

It is worth pointing out at this stage that this abstracts the long and short chain reduction into one reduction since in the case of the short chain it is already a pendant subtree in one of the trees. Additionally, because in rekernelising we wish to continually rekernelise, we also wish to have a chain reduction on a maximal chain that contains leaves that cross P. Fortunately this comes down to simply adding the respective values of the triples contained within the old chain to the new chain.

Lastly, as mentioned when calculating h' for multiple trees this chain reduction scales better, the number that needs to be added does not. The best way to do it in this case would be to save

the parent partition for the chain for each tree as the chain is extended. Then, when an element is removed from one end of the chain the other element is also removed and the weight added is the size of the smallest set, of whose elements each is a proper subset of all the sets in the parent partition. Although it is possible to have w keep track of all the different combinations—n trees needing 2^n associated numbers—this would scale poorly with the number of trees and is thus probably best avoided.

At this point one might wonder about using the two fairly different rekernelising algorithms for such closely related problems. Certainly with the following conjecture, and appropriate alterations to account for chain reduction, the bounded rooted subtree prune and regraft algorithm should also work for hybridisation.

Conjecture 4.27. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two phylogenetic trees and $\mathcal{F}$ a forest of $\mathcal{T}$ with no incompatible triples or overlapping components with respect to $\mathcal{T}'$. Let $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n\} \subseteq \mathcal{F}$ be a set of trees that are minimally cyclic in $G_{\mathcal{F}}$. Then in the maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ which is a subforest of $\mathcal{F}$ at least one of $\mathcal{T}_i$ for $i \in \{0, 1, \ldots, n\}$ has the edges adjacent to it's root removed breaking it into at least two pieces.*

The reason that the hybridisation rekernelising does not work for rooted subtree prune and regraft should be clear—we do not necessarily have vertices in $G_{\mathcal{F}}$ of out-degree zero. For the other consider that the rooted subtree prune and regraft rekernelising algorithm breaks up the forest based on which trees within the forest depend on which other trees. If we add the acyclic condition then every tree depends on all the trees that occur above and below it. We could cheat to a point and note the root is immune to this, however if this allows us to break up the forest then we have a common cluster on either side of the root so do not gain much. Now although the above conjecture could give a bounded search to the hybridisation problem given the stunning results obtained by hybrid rekernelising and the somewhat less stunning by bounded search for rooted subtree prune and regraft, which would be simpler than the hybridisation bounded search, it does not seem worthwhile to pursue this path.

4.4 Efficiently Deleting Edges Whilst Rekernelising

A way of picking leaves to be removed that will lead to a maximum agreement forest would be useful. Maybe someone else will determine such a result. At first it seemed as though one could

restrict the search to the leaves of incompatible triples, however figure 4.10 contains a counter example to this. Then it seemed possible the labels in the cluster of the root of the triple or the cluster in $\mathcal{T}'$ of the labels of the cluster in $\mathcal{T}'$ of the triple, but by this stage any benefit one may get out of it is looking small. We can consider the following result but in the end there is a much nicer result that actually runs more slowly when this result is used.

Lemma 4.28. *Let $\{a, b\}$ be a cherry in $\mathcal{T}$ and $\{b, c\}$ a cherry in $\mathcal{T}'$. Then at least one of a, b or c is a disconnected vertex in a maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$*

Proof. Trivial, taking $\mathcal{T} \mid \{a, b, c\}$ gives a $p_{\mathcal{T}|\{a,b,c\}}(a) = p_{\mathcal{T}|\{a,b,c\}}(b)$ and $\mathcal{T}' \mid \{a, b, c\}$ gives $p_{\mathcal{T}'|\{a,b,c\}}(b) = p_{\mathcal{T}'|\{a,b,c\}}(c)$, additionally since each of these parents is in a binary tree this completely describes their children. It then follows that one must be in a different label set to the others. It can then be shown that any one being in a label set of more than just itself necessitates detaching one of the others as an isolated vertex. □

Now we deal with the mixed blessing of rekernelising for there is a small gotcha. In the case of a brute force search in the previous algorithm there was a way to organise deletion of edges so that, even though each forest may be obtained multiple times, each possible deletion of edges is obtained only once. If our algorithm runs by detaching a pendant subtree then recursing, we expect runs of trees to be removed without any reductions and in this case evry permutation of these trees will be evaluated by the algorithm leading to a possibly longer run time as the results in Table 4.1 verify where the unaltered Interleaving technique is the column of results headed by *Rekernelising*. However, since we can assume there are some longish periods between reductions we can take advantage of this fact provided no other mechanisms for optimally selecting leaves are used. First a lemma.

Lemma 4.29. *There is a complete ordering on the members of an acyclic agreement forest $\mathcal{F}$ for a set of phylogenetic trees $\mathcal{P}$ as follows: if $\mathcal{T}_i$ needs to be removed for $\mathcal{T}_j$ to become pendant then $\mathcal{T}_i$ is less than $\mathcal{T}_j$ else if the label on the leaf, after reductions, in $\mathcal{T}_i$ is lower than the label on the leaf in $\mathcal{T}_j$ then $\mathcal{T}_i$ is less than $\mathcal{T}_j$.*

Proof. Let $\mathcal{F} = \{\mathcal{T}_\rho\}$. Clearly this forest is completely ordered. Assume the result holds for a forest of size k. Let $\mathcal{F}$ be a forest of size $k + 1$. Since $\mathcal{F}$ is acyclic there is some tree in $\mathcal{F}$ that is pendant in all members of $\mathcal{P}$ and contains the lowest leaf label in the trees, say $\mathcal{T}_A$. By the inductive hypothesis $\mathcal{F} - \{\mathcal{T}_A\}$ may be ordered with respect to $\mathcal{P}$ as described above. If

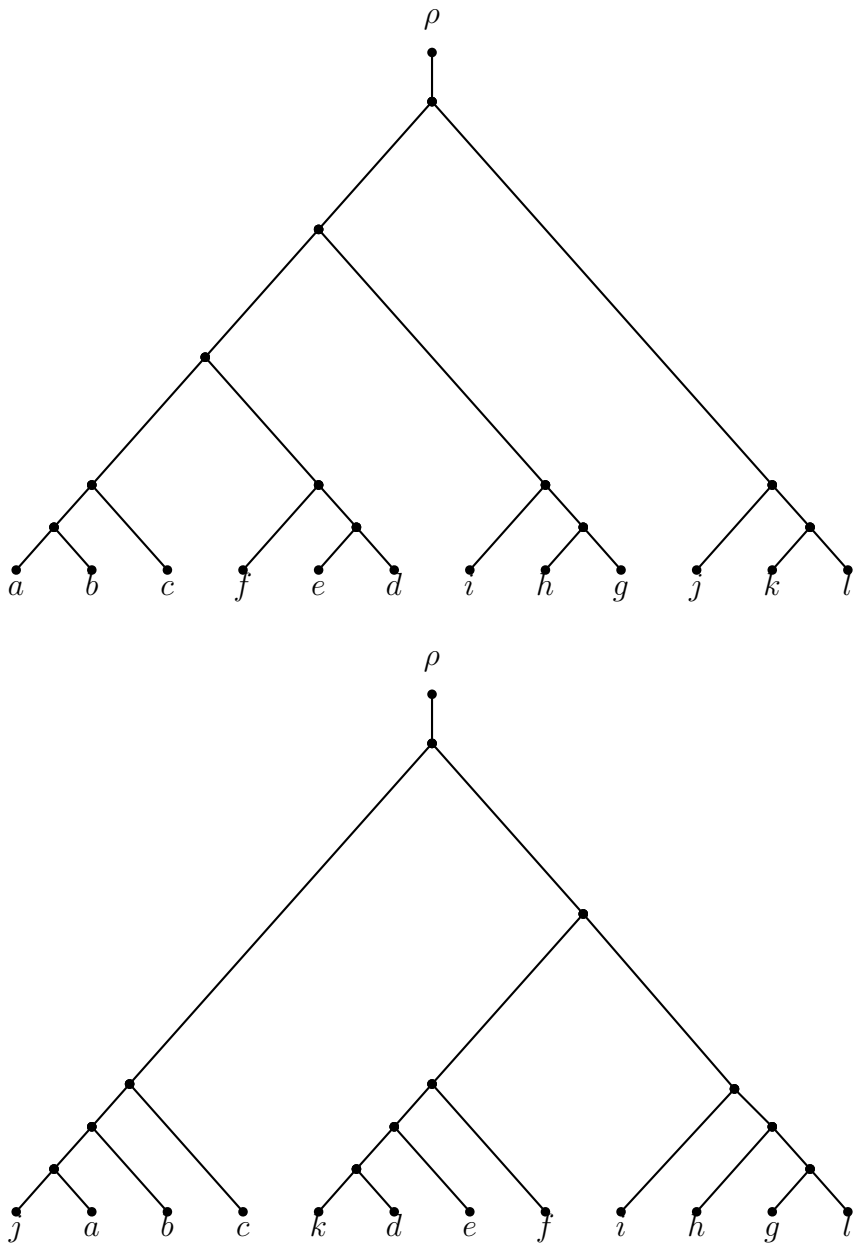


Figure 4.10: A pair of trees with no common subtrees where $ad|i$ is a minimal incompatible triple in $\mathcal{T}$ with respect to $\mathcal{T}'$ but where the maximum acyclic agreement forests are $\{\{j\},\{k\},\{l\},\{a,b,c,d,e,f,\rho\},\{g,h,i\}\}$, $\{\{j\},\{k\},\{l\},\{a,b,c,g,h,i,\rho\},\{f,e,d\}\}$ and $\{\{j\},\{k\},\{l\},\{d,e,f,g,h,i,\rho\},\{a,b,c\}\}$.

we consider the members of $\mathcal{P}$ as maximally reduced prior to the removal of $\mathcal{T}_A$ there are two subsequent possibilities. One is that they are still maximally reduced, in which case we have by definition that $\mathcal{T}_A$ is less than every member of the acyclic agreement forest as it contained the smallest leaf label. The other is that the members of $\mathcal{P}$ are no longer maximally reduced, in which case $\mathcal{T}_A$ is still less than every member of the acyclic agreement forest as it has a lower leaf label than all the members of the acyclic agreement forest that were not altered, and by definition comes before any members that may now be removed due to $\mathcal{T}_A$ being detached. □

The result above then allows us to build maximum acyclic agreement forests using the following, slightly weaker, result

1. The set of trees are reduced using subtree and chain reduction.
2. A pendant subtree is removed as per the interleaving algorithm.
3. If possible do reductions again.
4. If the reductions were possible then return to the beginning.
5. If the reductions were not possible then remove a tree with a label greater than the one previously removed. Then go to step 3.

The cluster reduction has been separated out because one can be cunning in dealing with it. Consider that if the subtree and chain reduction are not applicable prior to cluster reduction that the next leaf removed will be larger than the one previously removed. It then logically follows that if the cluster is detached the leaves removed may be larger than the one previously removed from the whole tree. However, since the tree from which the cluster was removed has undergone a reduction, the leaf one now removes from cannot be restricted. In Table 4.1 the interleaving algorithm that uses this result is the column headed by *Rekernelising*$++$.

Although this result is somewhat important in the sense that each maximum agreement forest will not be reached much more than once, it becomes extremely important when rekernelising as it removes many possible acyclic forests that are not maximal. Sadly it cannot be used unaltered in conjunction with the new non-binary chain reduction, as there may be removal of leaves that do not result in a reduction but which allow the chain to become pendant. If we keep the restriction on the available leaves to remove and have a chain whose members have the lowest labels of those available, then the chain will never have an opportunity to become

Grasses	#Taxa	h	Original	Weeding	Rekernelising	Rekernelising++
ndhF/phyt	40	14	5.9h	10.1m	21.5h	26.5s
ndhF/rbcL	36	13	5.3h	9.1m	1.5d	3.8s
ndhF/rpoC2	34	12	13.0h	20.4m	>2.0d	6.2s
ndhF/waxy	19	9	2.5m	2.0s	15.0s	0.3s
ndhF/ITS	46	19	>2.0d	>2.0d	>2.0d	4.3m
phyt/rbcL	21	4	0.7s	0.1s	0.0s	0.0s
phyt/rpoC2	21	7	1.5m	2.2s	5.3s	0.3s
phyt/waxy	14	3	0.2s	0.0s	0.0s	0.0s
phyt/ITS	30	8	9.9s	0.8s	0.3s	0.1s
rbcL/rpoC2	26	13	15.2h	38.4m	>2.0d	7.6s
rbcL/waxy	12	7	2.2m	1.7s	12.8s	0.5s
rbcL/ITS	29	14	>2.0d	>2.0d	>2.0d	10.2m
rpoC2/waxy	10	1	0.3s	0.0s	0.0s	0.0s
rpoC2/ITS	31	15	>2.0d	1.2d	>2.0d	57.4s
waxy/ITS	15	8	5.5m	5.5s	38.2s	0.3s

Table 4.1: Time taking for the implementation of various hybridisation number algorithms on the *poaceae* data set from Grass Phylogeny Working Group (2001)

pendant. This is not a problem in the binary case as either each element in the chain is isolated or the chain is contained within a subtree reduction. A hack that deals with this is to restrict the leaves under consideration to those larger than the one previously removed or any leaf that crosses P.

4.5 A Brief Introduction to Tree Bisection and Reconnection

At some point during the year spent doing Masters the question was posed to me *what is the hybridisation number of unrooted trees?*. The results discussed in previous *brief introductions...* have also often been considered for unrooted trees, with results of NP-hardness and fixed parameter tractability not rare. However, the question of subtree prune and regraft, whose definition follows closely from that of the rooted case, is significantly harder than the rooted case. How-

Algorithm 4.4.1: HYBRID-REKERNELISE$(\mathcal{T}, \mathcal{T}')$

procedure REDUCE$(\mathcal{T}, \mathcal{T}', w, k,$ `lastleafremoved`$)$
 if $k \leqslant 0$
  then return (k)
 `leafsetsize` $\leftarrow |\,\mathcal{L}(\mathcal{T})|$
 $(\mathcal{T}, \mathcal{T}') \leftarrow$ SUBTREEEREDUCTION$(\mathcal{T}, \mathcal{T}')$
 if $|\,\mathcal{L}(\mathcal{T})| \leqslant 3$
  then return (0)
 $(\mathcal{T}, \mathcal{T}', w) \leftarrow$ HYBRID-CHAINREDUCTION$(\mathcal{T}, \mathcal{T}', w)$
 if `leafsetsize` $\neq |\,\mathcal{L}(\mathcal{T})|$
  then `lastleafremoved` $\leftarrow 0$
 $C \leftarrow$ Labels of a minimal common cluster of $\mathcal{T}$ and $\mathcal{T}'$
 if $1 < |C| < |\,\mathcal{L}(\mathcal{T}')|$
  then $\begin{cases} (\mathcal{T}_1, \mathcal{T}_1', w_1, \mathcal{T}_2, \mathcal{T}_2', w_2) \leftarrow \text{CLUSTERREDUCTION}(\mathcal{T}, \mathcal{T}', w) \\ h \leftarrow \text{REKERNELISE}(\mathcal{T}_1, \mathcal{T}_1', w_1, k,\ \texttt{lastleafremoved}) \\ \textbf{return } (h + \text{REDUCE}(\mathcal{T}_2, \mathcal{T}_2', w_2, k-h, 0)) \end{cases}$
  else return (REKERNELISE$(\mathcal{T}, \mathcal{T}', w, k,$ `lastleafremoved`$)$)

procedure REKERNELISE$(\mathcal{T}, \mathcal{T}', w, k,$ `lastleafremoved`$)$
 for each $\ell \in \mathcal{L}(\mathcal{T}_1) - \{\rho_1\}$ **such that the label of** $\ell >$ **`lastleafremoved`**
  do $\begin{cases} \textbf{if } \exists \ell' \in \mathcal{L}(\mathcal{T}_1) - \{\rho_1, \ell\} \text{ such that } \{\ell, \ell'\} \in \operatorname{domain} w_1 \\ \quad \textbf{then} \begin{cases} \textbf{if } \ell > \ell' \\ \quad \textbf{then} \begin{cases} \mathcal{T} \leftarrow \mathcal{T} \mid \overline{\{\ell, \ell'\}} \\ \mathcal{T}' \leftarrow \mathcal{T}' \mid \overline{\{\ell, \ell'\}} \\ k \leftarrow \min\{k, \text{REDUCE}(\mathcal{T}, \mathcal{T}', w, k- \\ \qquad w(\{\ell, \ell'\}) - 2, \ell) + 2 + w(\{\ell, \ell'\})\} \end{cases} \end{cases} \\ \quad \textbf{else} \begin{cases} \mathcal{T}_1 \leftarrow \mathcal{T} \mid \overline{\{\ell\}} \\ \mathcal{T}_1' \leftarrow \mathcal{T}' \mid \overline{\{-\ell\}} \\ k \leftarrow \min\{k, \text{REDUCE}(\mathcal{T}_1, \mathcal{T}_1', w, k-1, \ell) + 1\} \end{cases} \end{cases}$
 return (k)

main
 $P \leftarrow \emptyset$
 $w : P \rightarrow \mathbb{Z}^+$
 $k \leftarrow$ REKERNELISE$(\mathcal{T}, \mathcal{T}', w, |\,\mathcal{L}(\mathcal{T})|, 0)$
 return (k)

ever the metric known as tree bisection and reconnection is able to use many of the results that have been obtained for rooted subtree prune and regraft and shall serve as a gentle introduction into how one deals with the unrooted case.

Let $\mathcal{T}$ be an unrooted phylogenetic tree on label set X whose interior vertices all have degree three. Let $e = \{u, v\}$ be an edge of $\mathcal{T}$ and $\mathcal{T}'$ to be the tree obtained by deleting e and attaching the component C_v that contains v to the component C_u that contains u via a new edge f. If f has an end in either u or v then $\mathcal{T}'$ has been obtained via a *(unrooted) subtree prune and regraft* or SPR operation. If we place no restriction on the new edge then $\mathcal{T}'$ has been obtained via a *tree bisection and reconnection* or TBR operation. The minimum number of the operations to transform $\mathcal{T}$ to $\mathcal{T}'$ induces a metric in both cases denoted $d_{\mathrm{SPR}}(\mathcal{T}, \mathcal{T}')$ and $d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}')$ respectively.

Theorem 4.30 (Theorem 2.4, Allen and Steel (2001))**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two binary phylogenetic trees with leaves labelled by X. Then*

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$$

Reading back to the original definitions it is easily observed that using maximum agreement forests in this fashion is unproblematic. The requirement was that they were non-overlapping subtrees of the trees under consideration, there was no requirement as to the direction of edges or even for a root. However, it is also the lack of direction on the edges that stops maximum agreement forests being used for subtree prune and regraft since if we take a tree in a maximum agreement forest for rooted trees then we can identify where it attached to the original tree by following the edges from head to tail until we can no longer, however in the unrooted, undirected case there is no way to indicate how the member of the forest was attached, which also makes it perfect for the tree bisection and reconnection distance.

Further, the minimum tree bisection and reconnection distance is NP-hard and is fixed parameter tractable.

Theorem 4.31 (Theorem 3.3, Allen and Steel (2001))**.** *The TBR distance problem is fixed parameter tractable.*

The reduction rules are identical to the ones used for the rooted subtree prune and regraft distance. Given this it then follows

Definition 4.32. *Let $\mathcal{T}$ and $\mathcal{T}$ be two phylogenetic trees with leaves labelled by X. Then*

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = \min_{\substack{\mathcal{B} \text{ is a binary refinement of } \mathcal{T} \\ \mathcal{B}' \text{ in a binary refinement of } \mathcal{T}'}} d_{\mathrm{TBR}}(\mathcal{B}, \mathcal{B}')$$

Further, we can make the characterisation

Corollary 4.33. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two phylogenetic trees with leaves labelled by X. Then*

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$$

Additionally it is easily shown that this problem is NP-hard, and fixed parameter tractable. This result follows easily from the considerations pointed out at the beginning of this section and the proofs in chapter 3.3

4.6 Unrooted Hybridisation

Biologists generally root a phylogenetic tree by including a species that is evolutionarily far away from those under consideration and then rooting the tree by that species. It seems likely, however, that there is data where this has not been done and where such tricks will not work should the entire gambit of life be under consideration. A natural question then becomes *is there a way of abstracting the hybridisation number to unrooted trees?* which in turn throws up a somewhat harder question of *what does it mean for a forest for an unrooted undirected tree to be acyclic*. With the disclaimer that I am not a biologist it seems logical to reason that, although we do not know where in the tree the root is, there must be one attached somewhere. With this in mind and the mantra that hybridisation events are unlikely define

$$h(\mathcal{T}, \mathcal{T}') = \min_{\substack{\underline{\mathcal{T}} \text{ is a rooting of } \mathcal{T} \\ \underline{\mathcal{T}'} \text{ is a rooting of } \mathcal{T}'}} h(\underline{\mathcal{T}}, \underline{\mathcal{T}'})$$

where

Definition 4.34. *$\underline{\mathcal{T}}$ is a* rooting *of a phylogenetic tree $\mathcal{T}$ if removing the root ρ from $\underline{\mathcal{T}}$ and contracting the edges of the resulting tree gives $\mathcal{T}$.*

The above definition is motivated by the idea that if one is presented with an unrooted phylogenetic tree then we could argue that there is a root attached *somewhere*. The following definition extends this by saying that if we are given a set of rooted phylogenetic trees then their acyclic agreement forests should also be acyclic agreement forests if they are unrooted.

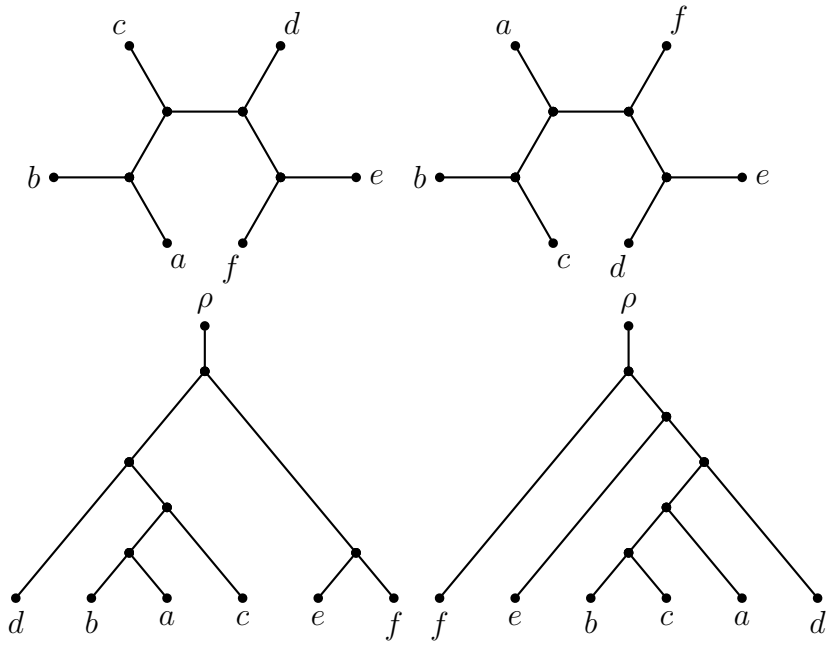


Figure 4.11: Two phylogenetic trees and two rootings which have the same acyclic agreement forest.

Definition 4.35. *A forest $\mathcal{F}$ is an acyclic agreement forest for a set of phylogenetic trees if it is an acyclic agreement forest for some rooting of each member of the set.*

Now the big result of this section. Given the definition for an acyclic agreement forest for an unrooted tree the question becomes how might one characterise it in order to make determining it easier. Magically if one uses the rekernelisation method to create an agreement forest for the trees it turns out that the forest constructed is acyclic as defined above. We shall extend the application and output of rekernelising a little to make it work however. In the unrooted trees consider the set consisting of all the maximal common subtrees and/or any leaves that are not members of said subtrees. Rekernelising shall, in this case, remove each of these in turn, rooting them at the point from which they used to be attached, and recurse. The resulting acyclic agreement forest shall have all but the final tree rooted.

Lemma 4.36. *For two unrooted phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with leaves labeled by X let $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n, \mathcal{T}_\rho\}$ be an acyclic agreement forest obtained by rekernelising, then there is a rooting $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}'}$ respectively such that $\{\mathcal{T}_1, \ldots, \mathcal{T}_n, \underline{\mathcal{T}_\rho}\}$, where $\underline{\mathcal{T}_\rho}$ is some rooting of $\mathcal{T}_\rho$, is an agreement forest for $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}'}$.*

Proof. If $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic than clearly taking any common edge and attaching a root gives rootings that have a hybridisation number of zero.

Assume it holds for agreement forests of size n. Let $\mathcal{T}$ and $\mathcal{T}'$ be unrooted phylogenetic trees such that rekernelising produces an agreement forest $\mathcal{F}$ of size $n+1$. Let A be the pendant subtree that rekernelising first removes and $\mathcal{T}_\rho$ the last tree that remains. Then $\mathcal{F} - \{A\}$ is an agreement forest for $\mathcal{T} \mid \overline{A}$ and $\mathcal{T}' \mid \overline{A}$ and by the inductive hypothesis there exist two rooted trees $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}}'$ such that $\mathcal{F} - \{A, \mathcal{T}_\rho\} \cup \{\underline{\mathcal{T}_\rho}\}$ is an acyclic agreement forest for the two trees. Next find the edge $e \in E(\mathcal{T})$ that is created from the contraction after the removal of A. Let $\mathcal{L}_1$ be the set of leaves on one side of the edge and $\mathcal{L}_2$ the set of leaves on the other side and $v \in \{\mathrm{mrca}_{\mathcal{T}} \mathcal{L}_1, \mathrm{mrca}_{\mathcal{T}} \mathcal{L}_2\} - \{\rho\}$. Subdivide the edge above v in $\mathcal{T}$ and attach A to the new vertex. Repeat the above for $\mathcal{T}'$ and $\underline{\mathcal{T}}'$. By construction $\mathcal{F} - \{\mathcal{T}_\rho\} \cup \{\underline{\mathcal{T}_\rho}\}$ is an acyclic agreement forest for $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}}'$. □

Lemma 4.37. *Let $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}}'$ be two rooted phylogenetic trees with an acyclic agreement forest $\mathcal{F}$ then $\mathcal{F}$ is also an acyclic agreement forest for the respective unrooted trees $\mathcal{T}$ and $\mathcal{T}'$.*

Proof. Let $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}}'$ be isomorphic, then $\mathcal{F} = \{\mathcal{T}_\rho\}$ is an acyclic agreement forest for the trees and $\mathcal{F} = \{\mathcal{T}_\rho \mid \overline{\{\rho\}}\}$ is an acyclic agreement forest for the unrooted trees.

Assume the lemma is true for forests of size n. Let $\underline{\mathcal{T}}$ and $\underline{\mathcal{T}}'$ be trees with an acyclic agreement forest of size $n+1$. Since the forest is acyclic there is at least one vertex in $G_{\mathcal{F}}$ with out-degree zero which is a pendant subtree in both trees—let this tree be A. $\mathcal{F} - \{A\}$ will be an acyclic agreement forest for $\underline{\mathcal{T}} \mid \overline{A}$ and $\underline{\mathcal{T}}' \mid \overline{A}$ and thus by the inductive hypothesis $\mathcal{F} - \{A, \mathcal{T}_\rho\} \cup \{\mathcal{T}_\rho \mid \overline{\{\rho\}}\}$ is an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Find the sets $\mathcal{L}_1 = \mathcal{C}(\mathrm{mrca}_{\underline{\mathcal{T}}} A) - A$ and $\mathcal{L}_1 = X - \mathcal{C}(\mathrm{mrca}_{\underline{\mathcal{T}}} A)$. In $\mathcal{T}$ find the edge e in $\mathcal{T}$ connecting $\mathcal{T}(\mathcal{L}_1)$ and $\mathcal{T}(\mathcal{L}_2)$. Subdivide e and attach A to the new vertex. Repeat the above steps to $\mathcal{T}'$ then by construction $\mathcal{F}$ is an acyclic agreement forest for $\mathcal{T}$ □

So we have the characterisation for maximum acyclic agreement forests for unrooted trees via the construction of forests that utilises the rekernelisation algorithm. Ideally the problem could be shown to be fixed parameter tractable. The subtree reduction essentially drops out of the proofs above. A cluster reduction is detailed below. However, in the papers published thus far to have a chain reduction we need a direction on the edges, which we no longer usually have. So although there are reductions available it is no longer clear that this problem is fixed parameter tractable.

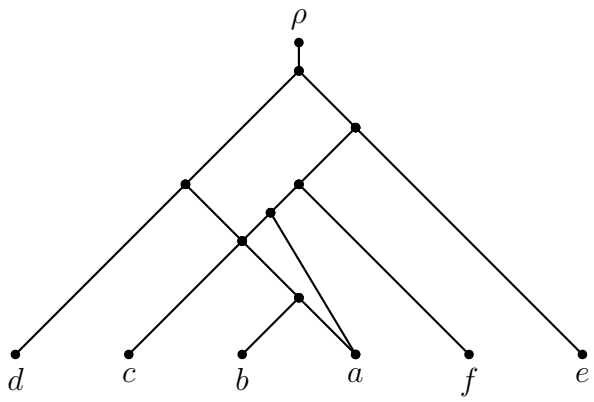


Figure 4.12: A phylogenetic network of with the fewest number of hybridisation vertices for the unrooted trees in figure 4.11.

Lemma 4.38. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unrooted phylogenetic trees with a common cluster A, noting then that $X - A$ is also a common cluster. Let $\mathcal{T}_a$ and $\mathcal{T}'_a$ be the cluster reduced tree-pair obtained from $\mathcal{T}$ and $\mathcal{T}'$ by applying the cluster reduction to A and suppose that $\mathcal{L}(\mathcal{T}_a) = \mathcal{L}(\mathcal{T}'_a) = (X - A) \cup \{a\}$ with $a \notin X$. Similarly let $\mathcal{T}_b$ and $\mathcal{T}'_b$ be the cluster reduced tree-pair obtained from $\mathcal{T}$ and $\mathcal{T}'$ by applying the cluster reduction to $X - A$ and suppose that $\mathcal{L}(\mathcal{T}_b) = \mathcal{L}(\mathcal{T}'_b) = A \cup \{b\}$ with $b \notin X$. Then, noting that $\mathcal{T} \mid A$, $\mathcal{T}' \mid A$, $\mathcal{T} \mid \overline{A}$ and $\mathcal{T}' \mid \overline{A}$ are rooted trees whereas $\mathcal{T}_a$, $\mathcal{T}'_a$, $\mathcal{T}_b$ and $\mathcal{T}'_b$ are unrooted trees,*

$$h(\mathcal{T}, \mathcal{T}') = \min\left\{h(\mathcal{T}_a, \mathcal{T}'_a) + h(\mathcal{T} \mid A, \mathcal{T}' \mid A), h(\mathcal{T}_b, \mathcal{T}'_b) + h(\mathcal{T} \mid \overline{A}, \mathcal{T}' \mid \overline{A})\right\}$$

Proof. In the arbitrary rootings there are a few possibilities. Let $\rho_{\mathcal{T}}$ be the root that is attached to $\mathcal{T}$ and $\rho_{\mathcal{T}'}$ be the root that is attached to $\mathcal{T}'$ to obtain a maximum acyclic agreement forest then enumerating all the possibilities

1. $\rho_{\mathcal{T}}$ and $\rho_{\mathcal{T}'}$ are attached via an edge to the parts of $\mathcal{T}$ and $\mathcal{T}'$ corresponding to $\mathcal{T}_a$ and $\mathcal{T}'_a$ respectively or $\mathcal{T}_b$ and $\mathcal{T}'_b$ respectively.

2. $\rho_{\mathcal{T}}$ and $\rho_{\mathcal{T}'}$ are attached via an edge to the parts of $\mathcal{T}$ and $\mathcal{T}'$ corresponding to $\mathcal{T}_a$ and $\mathcal{T}'_b$ respectively or $\mathcal{T}_b$ and $\mathcal{T}'_a$ respectively.

3. $\rho_{\mathcal{T}}$ and $\rho_{\mathcal{T}'}$ are attached via an edge to the edge that separates A and $X - A$ in $\mathcal{T}$ and $\mathcal{T}'$ respectively.

In the first case the result is trivially true as either A or $X - A$ becomes a pendant cluster in both trees. In the last case then both of the clusters are rooted trees. It then follows that if this gives

a maximum acyclic agreement forest that lemma's statement will too, as they will be rootings of $\mathcal{T}_a$, $\mathcal{T}'_a$, $\mathcal{T}_b$ or $\mathcal{T}'_b$.

Let $\mathcal{F}$ be an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ and $\mathcal{T}_\rho$ be the final tree that rekernelising removes. We know that for a maximum acyclic agreement forest ρ will not be isolated. This follows from the lemma in Linz (2008) which states for a pair of rooted binary phylogenetic trees ρ is never isolated. Since we are taking the minimum of all binary representations of all rootings of phylogenetic trees it follows that ρ will not be isolated in any of these. Further, consider that no matter where we attach the root in the rooting for $\mathcal{T}_\rho$ the already removed trees that make up the rest of $\mathcal{F}$ will not have their rootings altered. For the root to be attached to a tree that straddles A and $X - A$ we must have $A \cap \mathcal{L}(\mathcal{T}_\rho) \neq \emptyset$ and $(X - A) \cap \mathcal{L}(\mathcal{T}_\rho) \neq \emptyset$. For the resulting rooting to be isomorphic we must have the root either adjacent to an edge in $\mathcal{T}_a$ and $\mathcal{T}'_a$ or $\mathcal{T}_b$ and $\mathcal{T}'_b$, which we have already stated not to be the case, or adjacent to the edge that connects the two sets. If it adjacent to this edge then we may move the root from A to $X - A$ or vice versa without altering the acyclic agreement forest which brings us back to the first situation. □

Chapter 5

Future Work

> Hofstadter's Law: "It always takes longer than you expect, even when you take Hofstadter's Law into account."
>
> (Hofstadter, 1979)

Relatively early in the game a request was placed that I write code which gave output a phylogenetic network in extended newick for an agreement forest for a pair of trees. Relatively late in the game, when results could really be obtained quickly, my attention turned here. Part of this was that one could return *a* phylogenetic network or *all* phylogenetic networks. Although there is a published algorithm that, given an acyclic agreement forest and a set of binary phylogenetic trees will construct a phylogenetic network in polynomial time (Baroni et al., 2006), in the superficial amount of time I spent thinking about it there did not seem to be a clear way to improve it to give all phylogenetic networks, or to work comfortably with non-binary trees. At present I have software that was being worked on prior to crunch time that, if given a set of not necessarily binary phylogenetic trees, will output a or all maximum acyclic agreement forests (however the no chain reduction is being used as yet).

When proving the results about the reductions for the rekernelisation algorithm for rooted subtree prune and regraft there seemed to be a result that would result in a stronger chain reduction.

Conjecture 5.1. *Let $\mathcal{F}$ be a forest and $\mathcal{T}$ a phylogenetic tree. Let A be a set such that $\mathcal{T} \mid A$ is a pendant subtree of $\mathcal{T}$ and such that the forest $\mathcal{F}' = \{\mathcal{T}_i \mid A : \mathcal{T}_i \in \mathcal{F}\}$ has the property that $|\mathcal{F}' - \mathcal{F}| = 1$ and $\mathcal{T}' \in \mathcal{F}' - \mathcal{F}$ is a pendant subtree for some tree in $\mathcal{F}$. If $\mathcal{F}'$ exists then attach*

a root to $\mathcal{T}_i$ and

$$d_{\mathrm{rSPR}}(\mathcal{F},\mathcal{T}) = d_{\mathrm{rSPR}}(\mathcal{F}',\mathcal{T} \mid A \cup \{\rho\}) + d_{\mathrm{rSPR}}(\mathcal{F} \mid \overline{A},\mathcal{T} \mid \overline{A})$$

if there is an agreement forest for $\mathcal{F}'$ and $\mathcal{T} \mid A \cup \{\rho\}$ with ρ isolated, otherwise

$$d_{\mathrm{rSPR}}(\mathcal{F},\mathcal{T}) = d_{\mathrm{rSPR}}(\mathcal{F}',\mathcal{T} \mid A \cup \{\rho\}) + d_{\mathrm{rSPR}}(\mathcal{F} \mid \overline{A - \min A},\mathcal{T} \mid \overline{A - \min A})$$

This would also lead to a stronger subtree reduction, the latter being a special case of the former. Also in the rooted subtree prune and regraft arena it seems that a bounded search for non-binary rooted subtree prune and regraft should exist.

As mentioned earlier in this thesis it appears as though both rekernelising algorithms would lend themselves well to multiprocessing. Results along these lines will become increasingly important as multicore PC's become prevalent thus a reworking and implementation of the algorithms in this thesis to be multithreaded seems like a useful and not overly difficult task to be dealt to.

Bibliography

Benjamin L. Allen and Mike Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:2001, 2001. http://www.math.canterbury.ac.nz/~m.steel/Non_UC/files/research/rearrangement.pdf.

Mihaela Baroni, Charles Semple, and Mike Steel. A framework for representing reticulate evolution. *ACOM*, 8:398–401, 2004. http://www.math.canterbury.ac.nz/~c.semple/papers/BSS04.pdf.

Mihaela Baroni, Stefan Grünewald, Vincent Moulton, and Charles Semple. Bounding the number of hybridisation events for a consistent evolutionary history. *Journal of Mathematical Biology*, 51:171–182, 2005. http://www.math.canterbury.ac.nz/~c.semple/papers/BGMS05.pdf.

Mihaela Baroni, Charles Semple, and Mike Steel. Hybrids in real time. *Systematic Biology*, 55:46–56, 2006. http://www.math.canterbury.ac.nz/~c.semple/papers/BSS06.pdf.

Robert G. Beiko and Nicholas Hamilton. Phylogenetic identification of lateral genetic transfer events. *BMCEB*, 6(15), 2006. http://dx.doi.org/10.1186/1471-2148-6-15.

Magnus Bordewich and Charles Semple. On the computational complexity of the rooted subtree prune and regraft distance. *Annals of Combinatorics*, 8:409–423, 2004. http://www.math.canterbury.ac.nz/~c.semple/papers/BS04.pdf.

Magnus Bordewich and Charles Semple. Computing the minimum number of hybridization events for a consistent evolutionary history. *DAM*, 155:914–918, 2007a. http://www.math.canterbury.ac.nz/~c.semple/papers/BS06a.pdf.

Magnus Bordewich and Charles Semple. Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable. *TCBB*, 4(3):458–466, 2007b. http://www.math.canterbury.ac.nz/~c.semple/papers/BS06b.pdf.

Magnus Bordewich, Simone Linz, Katherine St. John, and Charles Semple. A reduction algorithm for computing the hybridization number of two trees. *EBIO*, 3:86–98, 2007a. http://www.math.canterbury.ac.nz/~c.semple/papers/BLSS07.pdf.

Magnus Bordewich, Catherine McCartin, and Charles Semple. A 3-approximation algorithm for the subtree distance between phylogenies. *Journal of Discrete Algorithms*, 2007b. http://www.math.canterbury.ac.nz/~c.semple/papers/BMS07.pdf.

David Bryant and Vincent Moulton. Neighbornet: An agglomerative method for the construction of phylogenetic networks. *MBE*, 21(2):255–265, 2004. http://www.math.auckland.ac.nz/~bryant/Papers/04NeighborNet.pdf.

Joshua Collins, Simone Linz, and Charles Semple. Quantifying hybridization in realistic time. *Systematic Biology*, submitted.

Charles Darwin. *The Origin of Species*. John Murray, 1859. http://publicliterature.org/books/origin_of_species/xaa.php.

W. F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284:2124–2128, 1999. bioinfo.mbi.ucla.edu/courses/m298/Doolittle%20wf.pdf.

R. Downey and M. Fellows. *Parameterized Complexity*. Springer Publishing, 1998.

J. Flum and G. Grohe. *Parameterized Complexity Theory*. Springer Publishing, 2006.

V. A. Funk. Phylogenetic patterns and hybridization. *Annals of the Missouri Botanical Garden*, 72:681–715, 1985. http://www.jstor.org/stable/2399220.

Grass Phylogeny Working Group. Phylogeny and subfamilial classification of the grasses (*Poaceae*). 88:373–457, 2001.

Dan Gusfield and Vikas Bansal. A fundamental decomposition theory for phylogenetic networks and incompatible characters. In *RECOMB05*, volume 3500 of *LNCS*,

pages 217–232. springer, 2005. http://wwwcsif.cs.ucdavis.edu/~gusfield/gusfieldrecomb.pdf.

Dan Gusfield, Satish Eddhu, and Charles Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *JBCB*, 2(1):173–213, 2004. http://wwwcsif.cs.ucdavis.edu/~gusfield/exfinalrec.pdf.

Mike Hallett and Jens Lagergren. Efficient algorithms for lateral gene transfers problems, 2004. submitted to SIAM Journal on Computing, http://www.mcb.mcgill.ca/~hallett/Lateral.pdf.

Jotun Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *MBIO*, 98(2):185–200, 1990. http://dx.doi.org/10.1016/0025-5564(90)90123-G.

Jotun Hein, Tao Jiang, Lusheng Wang, and Jaizhong Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, 71:153–169, 1996. http://www.sciencedirect.com/science/article/B6TYW-3VTK33B-9/2/9a02c096ea03891a35e2c2a66800d6dc.

Douglas R. Hofstadter. *Gödel Escher Bach*. Basic Books, 1979.

Barbara R. Holland, Katharina T Huber, Vincent Moulton, and Peter J. Lockhart. Using consensus networks to visualize contradictory evidence for species phylogeny. *MBE*, 21(7):1459–1461, 2004. http://dx.doi.org/10.1093/molbev/msh145.

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley, 2 edition, 2001.

Daniel H. Huson, Tobias Dezulian, Tobias Kloepper, and Mike Steel. Phylogenetic supernetworks from partial trees. *TCBB*, 1(4):151–158, 2004. http://awcmee.massey.ac.nz/pdf_files/Huson_et_al_2004.pdf.

Daniel H. Huson, Tobias Kloepper, Peter J. Lockhart, and Mike Steel. Reconstruction of reticulate networks from gene trees. In *RECOMB05*, volume 3500 of *LNCS*, pages 233–249. springer, 2005. http://dx.doi.org/10.1007/11415770_18.

Pierre Legendre and Vladimir Makarenkov. Reconstruction of biogeographic and evolutionary networks using reticulograms. *SB*, 51(2):199–216, 2002. http://www.info.uqam.ca/~makarenv/articles/article_SB.pdf.

Simone Linz. *Reticulation in evolution*. PhD thesis, Heinrich-Heine-University, Düsseldorf, Germany, 2008. http://www.cs.uni-duesseldorf.de/NewMA/Personen/entry_43/Dokument e/linzPhD.pdf.

Simone Linz and Charles Semple. Hybridization in non-binary trees, 2008. Submitted to TCBB, slides available at http://www.newton.cam.ac.uk/webseminars/pgws/2007/plg/plgw03/1220/linz.

Simone Linz and Charles Semple. A cluster reduction for computing the subtree distance between phylogenies. *Annals of Combinatorics*, in press.

David R. Maddison. The discovery and importance of multiple islands of most-parsimonious trees. *Systematic Zoology*, pages 315–328, 1991. http://www.jstor.org/stable/2992325.

Wayne P. Maddison. Gene trees in species trees. *SB*, 46(3):523–536, 1997. http://dx.doi.org/10.2307/2413694.

Geoff McFadden and Paul Gilson. Something borrowed, something green: lateral transfer of chloroplasts by secondary endosymbiosis. *Trends in Ecology & Evolution*, pages 12–17, 1995. http://www.sciencedirect.com/science/article/B6VJ1-40W0SGG-6/2/44fd9b345389b2d15e31d53d084126bf.

Bernard M. E. Moret, Luay Nakhleh, Tandy Warnow, C. Randal Linder, Anna Tholse, Anneke Padolina, Jerry Sun, and Ruth Timme. Phylogenetic networks: Modeling, reconstructibility, and accuracy. *TCBB*, 1(1):13–23, 2004. http://www.cs.rice.edu/~nakhleh/Papers/tcbb04.pdf.

Luay Nakhleh, Don Ringe, and Tandy Warnow. Perfect phylogenetic networks: A new methodology for reconstructing the evolutionary history of natural languages. *Language, Journal of the Linguistic Society of America*, 81(2):382–420, 2002. http://www.cs.rice.edu/~nakhleh/Papers/81.2nakhleh.pdf.

Luay Nakhleh, Tandy Warnow, C. Randal Linder, and Katherine St. John. Reconstructing reticulate evolution in species - theory and practice. *JCB*, 12(6):796–811, 2005. http://www.cs.rice.edu/~nakhleh/Papers/NWLSjcb.pdf.

R. Niedermeier and P. Rossmanith. A general method to speed up fixed-parameter-tractable algorithms. *Inform. Process. Lett.*, (3):1545–5963, 2000.

David Posada and Keith A. Crandall. Intraspecific gene genealogies: trees grafting into networks. *TEE*, pages 37–45, 2001. http://darwin.uvigo.es/download/papers/09.networks01.pdf.

Estela Maris Rodrigues, Marie-France Sagot, and Yoshiko Wakabayashi. *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, chapter Some Approximation Results for the Maximum Agreement Forest Problem, pages 159–169. Springer Berlin / Heidelberg, 2001. http://www.springerlink.com/content/t5da4wxaeact3dg8/.

C. Semple and M. Steel. *Phylogenetics*. Oxford Univerisity Press, 2003.

Charles Semple. Hybridization networks. In Olivier Gascuel and Mike Steel, editors, *Reconstructing Evolution, New Mathematical and Computational Advances*, pages 277–314. Oxford University Press, 2007. http://www.math.canterbury.ac.nz/~c.semple/papers/S06.pdf.

Yun S. Song and Jotun Hein. Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events. In *WABI03*, volume 2812 of *LNCS*, pages 287–302. springer, 2003. http://www.cs.ucdavis.edu/~yssong/Pub/SH-WABI03.pdf.

Yun S. Song and Jotun Hein. Constructing minimal ancestral recombination graphs. *JCB*, 12(2):147–169, 2005. http://mathgen.stats.ox.ac.uk/bioinformatics/publications/songhein.5.JCB-12-147.pdf.

Matthew Spencer, Elizabeth A. Davidson, Adrian C. Barbrook, and Christopher J. Howe. Phylogenetics of artificial manuscripts. *Journal of Theoretical Biology*, pages 503–511, 2004.

List of Figures

List of Tables

List of Algorithms

Index